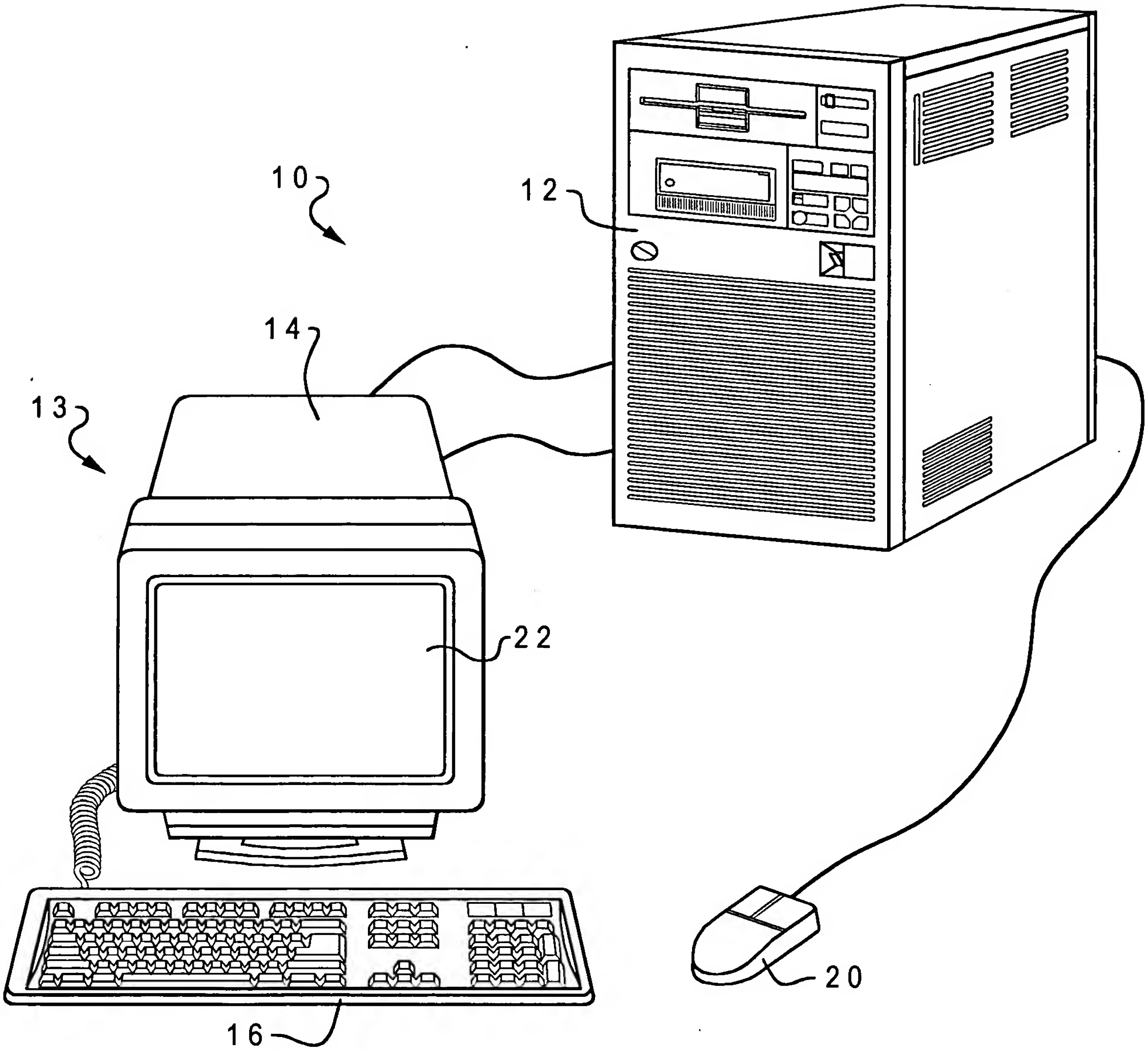


Fig. 1

Prior Art

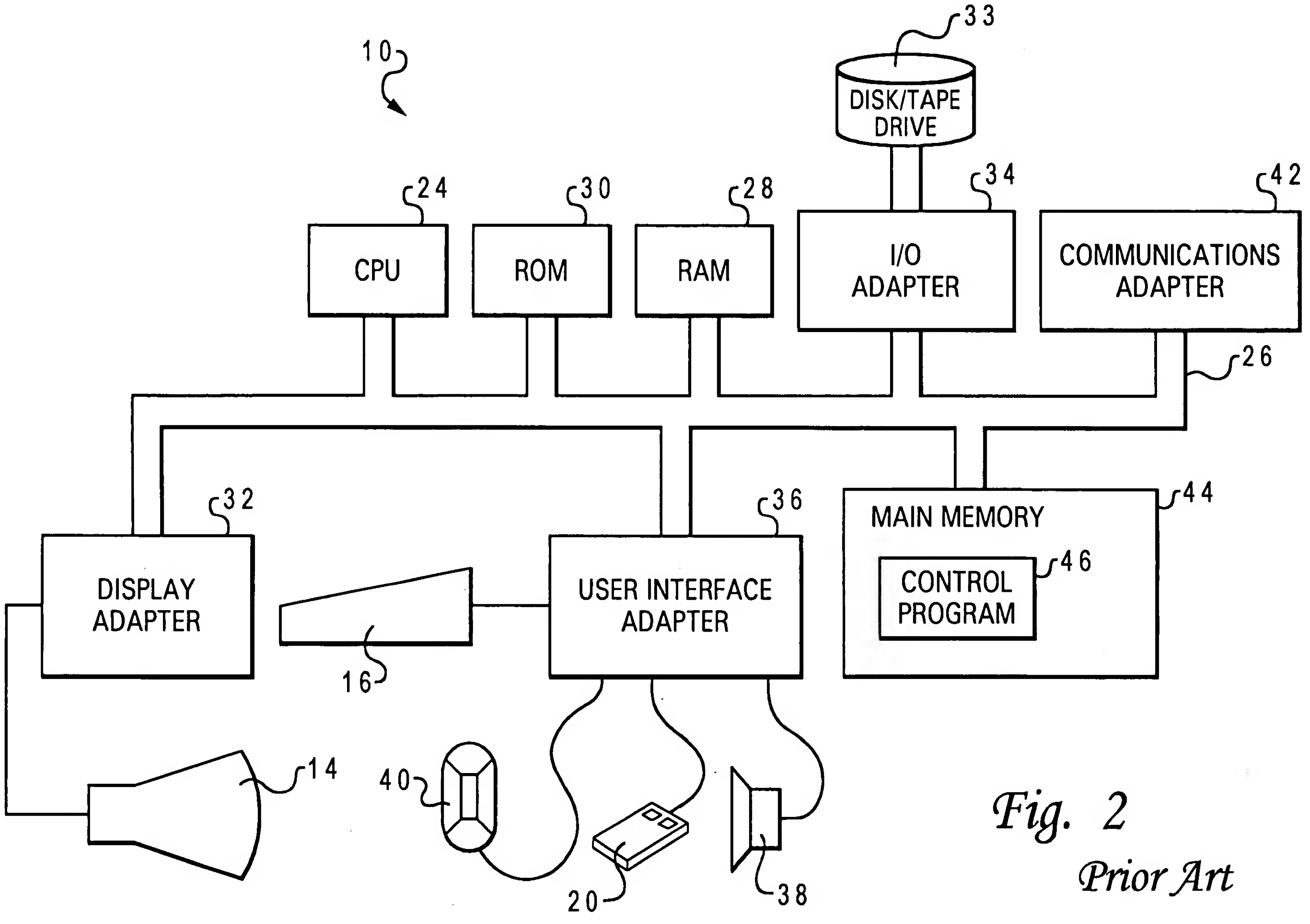


Fig. 2

Prior Art

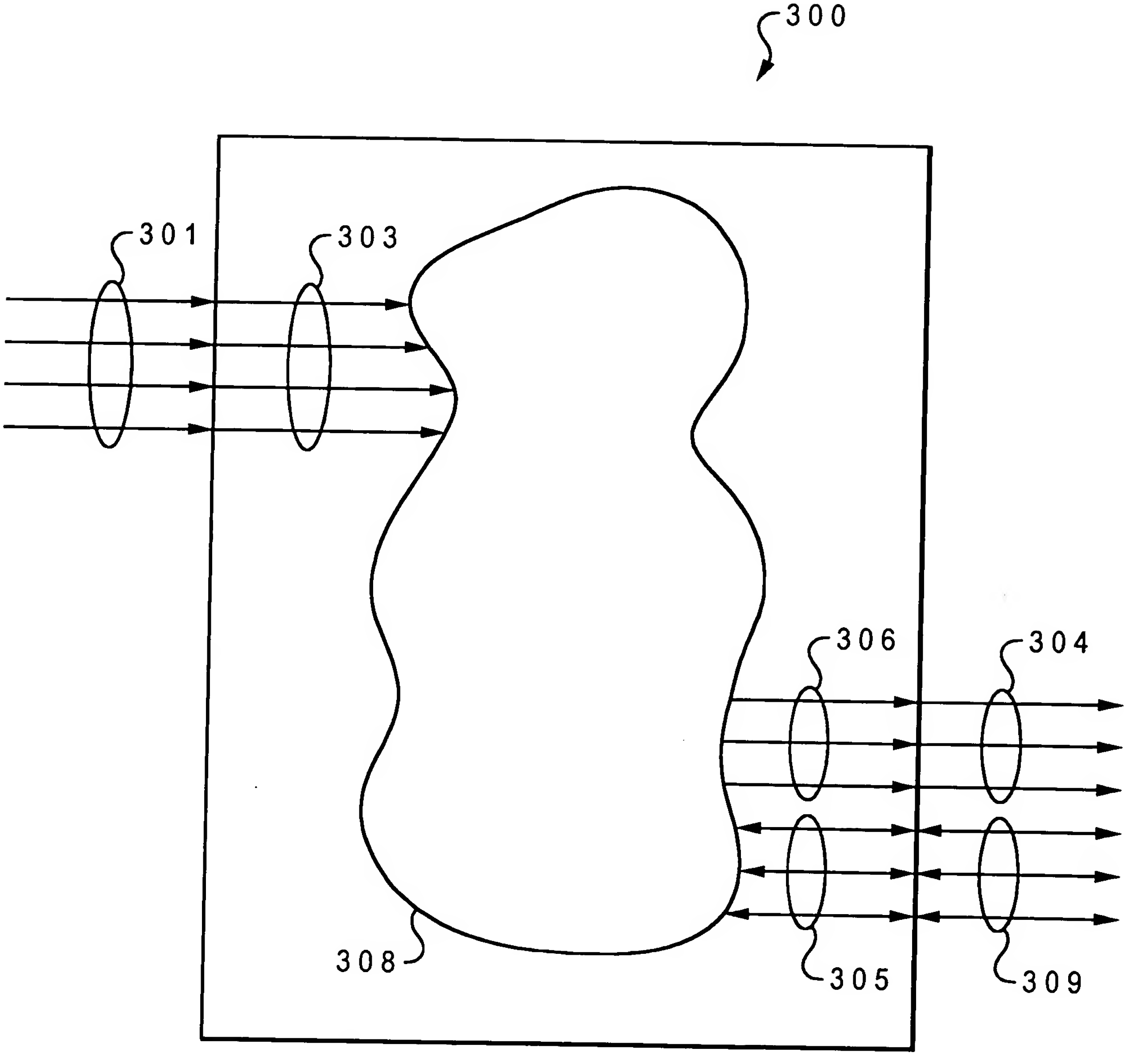


Fig. 3A

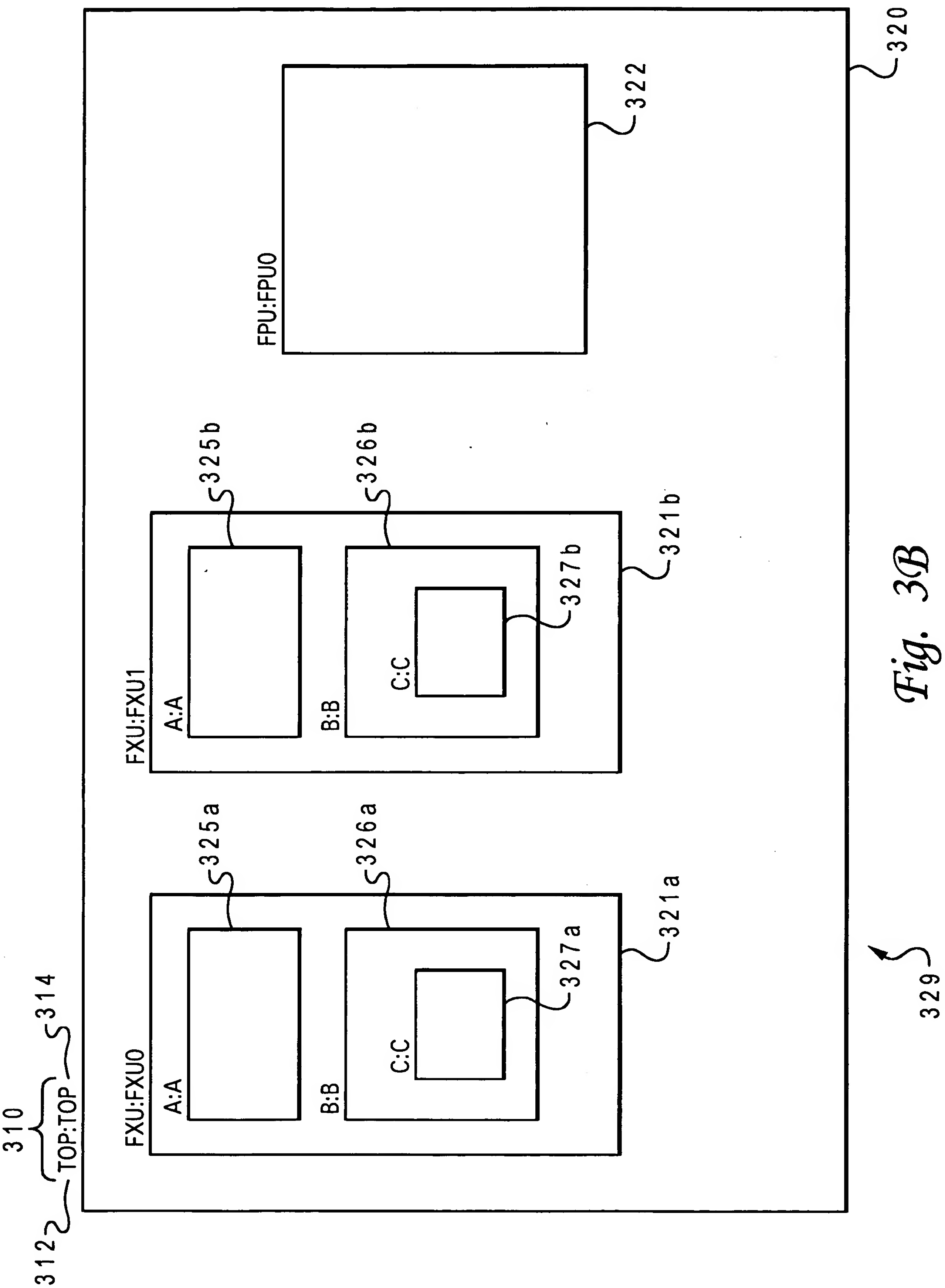

310
TOP:TOP
312
314
FXU:FXU0
A:A
B:B
C:C
325a
326a
327a
321a
FXU:FXU1
A:A
B:B
C:C
325b
326b
327b
321b
FPU:FPU0
322
320
329

Fig. 3B

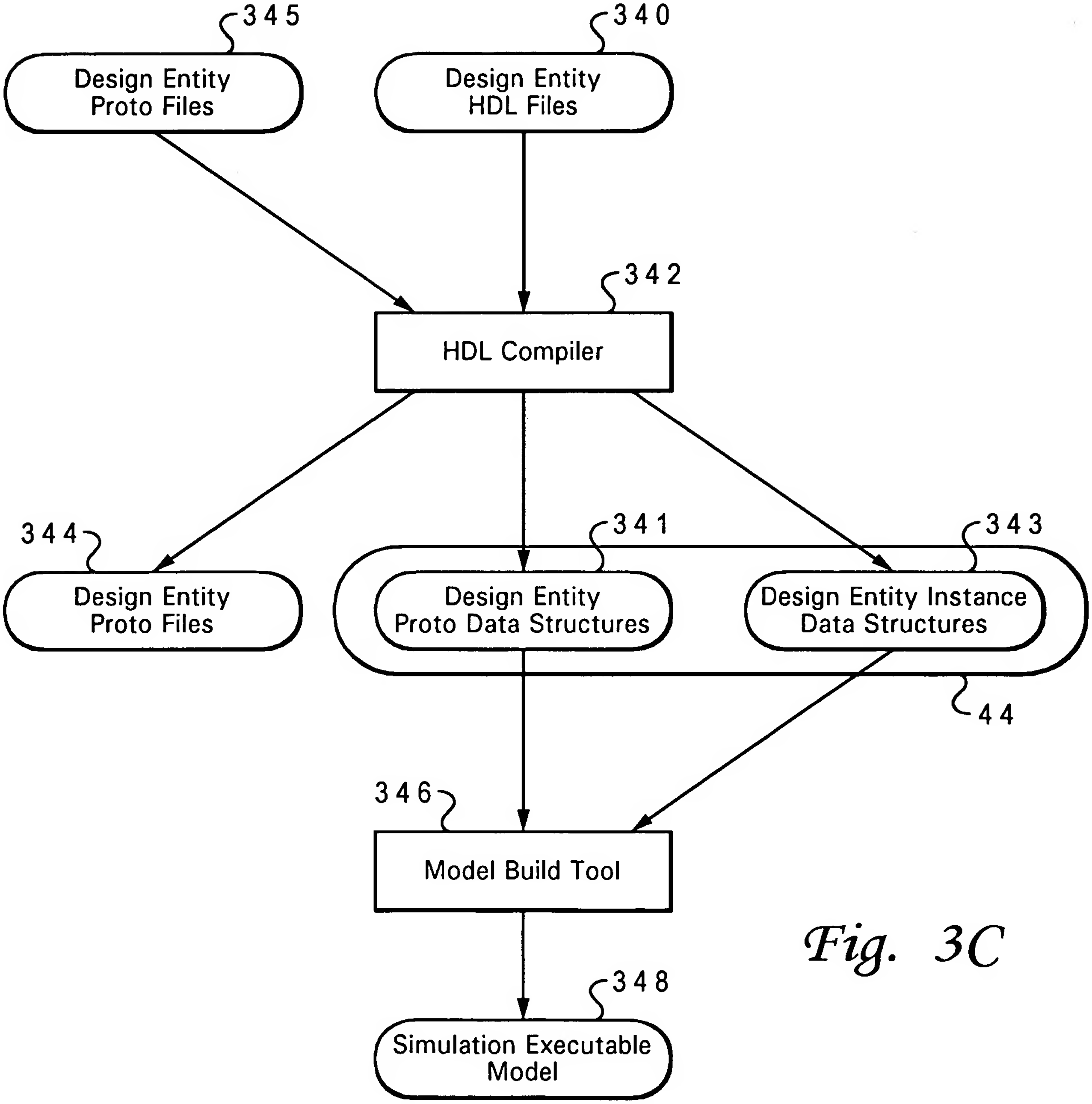

345
Design Entity Proto Files
340
Design Entity HDL Files
342
HDL Compiler
344
Design Entity Proto Files
341
Design Entity Proto Data Structures
343
Design Entity Instance Data Structures
44
346
Model Build Tool
348
Simulation Executable Model

Fig. 3C

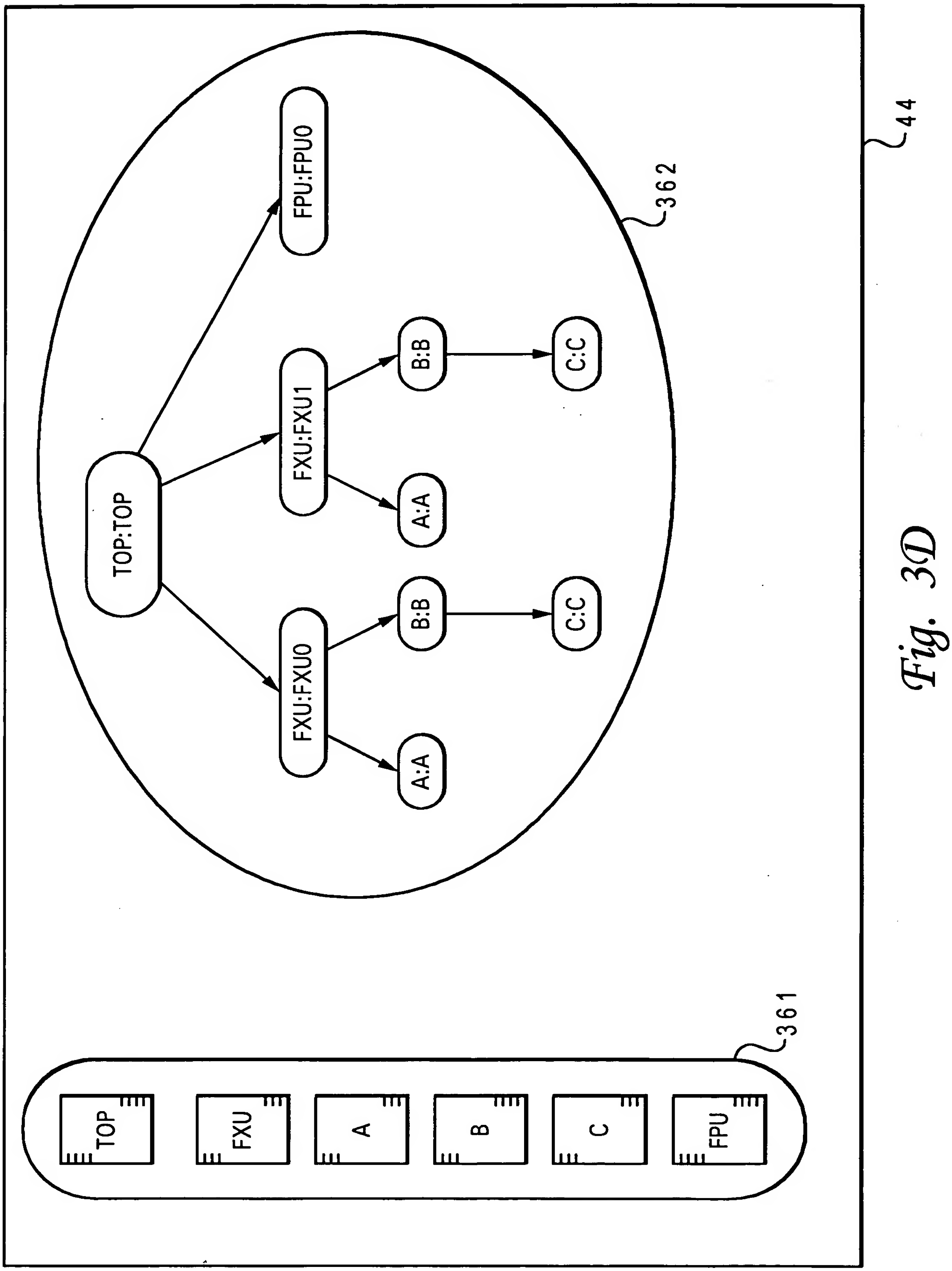

44
362
TOP:TOP
FXU:FXU0
FXU:FXU1
FPU:FPU0
A:A
B:B
C:C
A:A
B:B
C:C
361
TOP
FXU
A
B
C
FPU

Fig. 3D

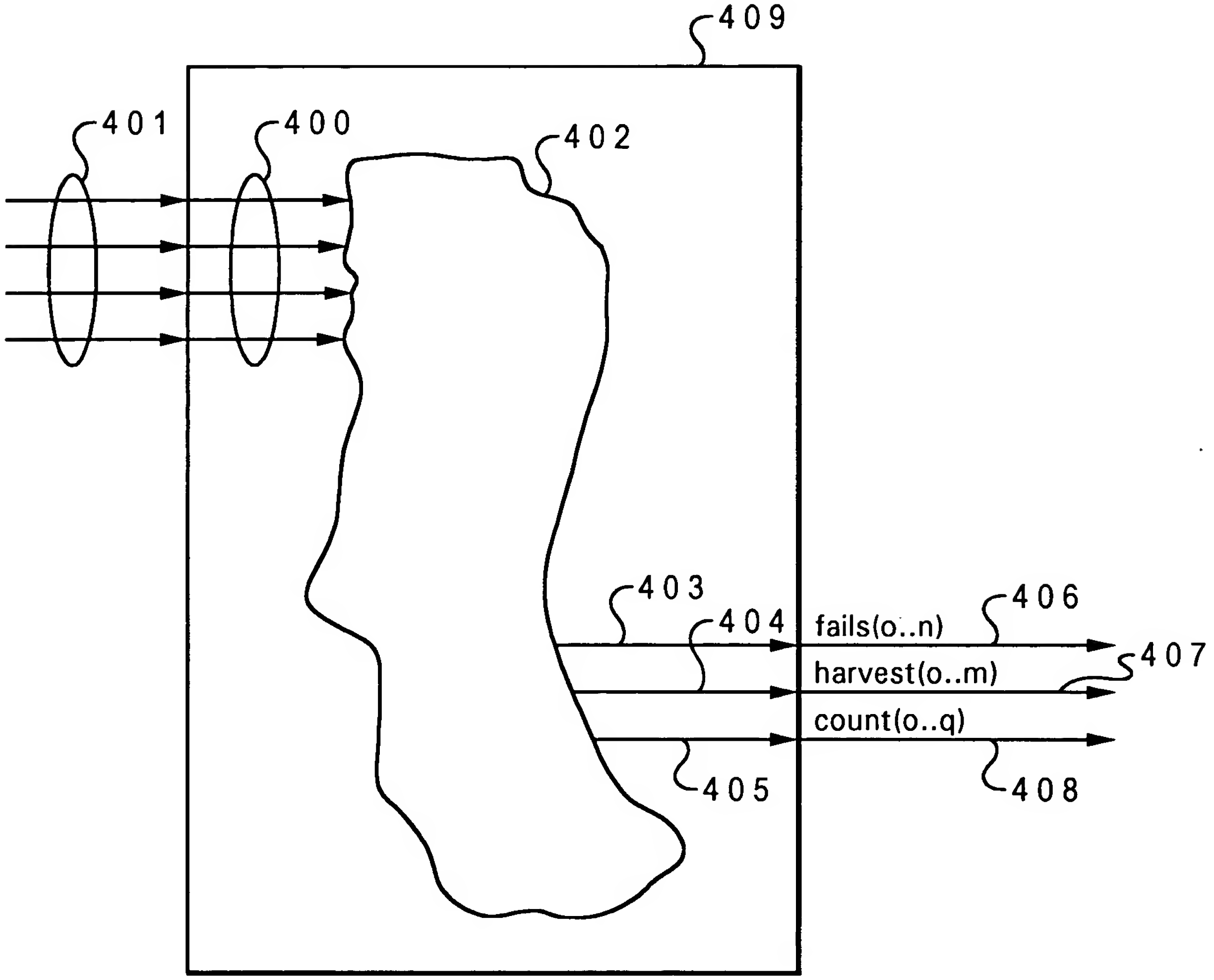

409
401
400
402
403
404
fails(o..n)
406
407
harvest(o..m)
count(o..q)
405
408

Fig. 4A

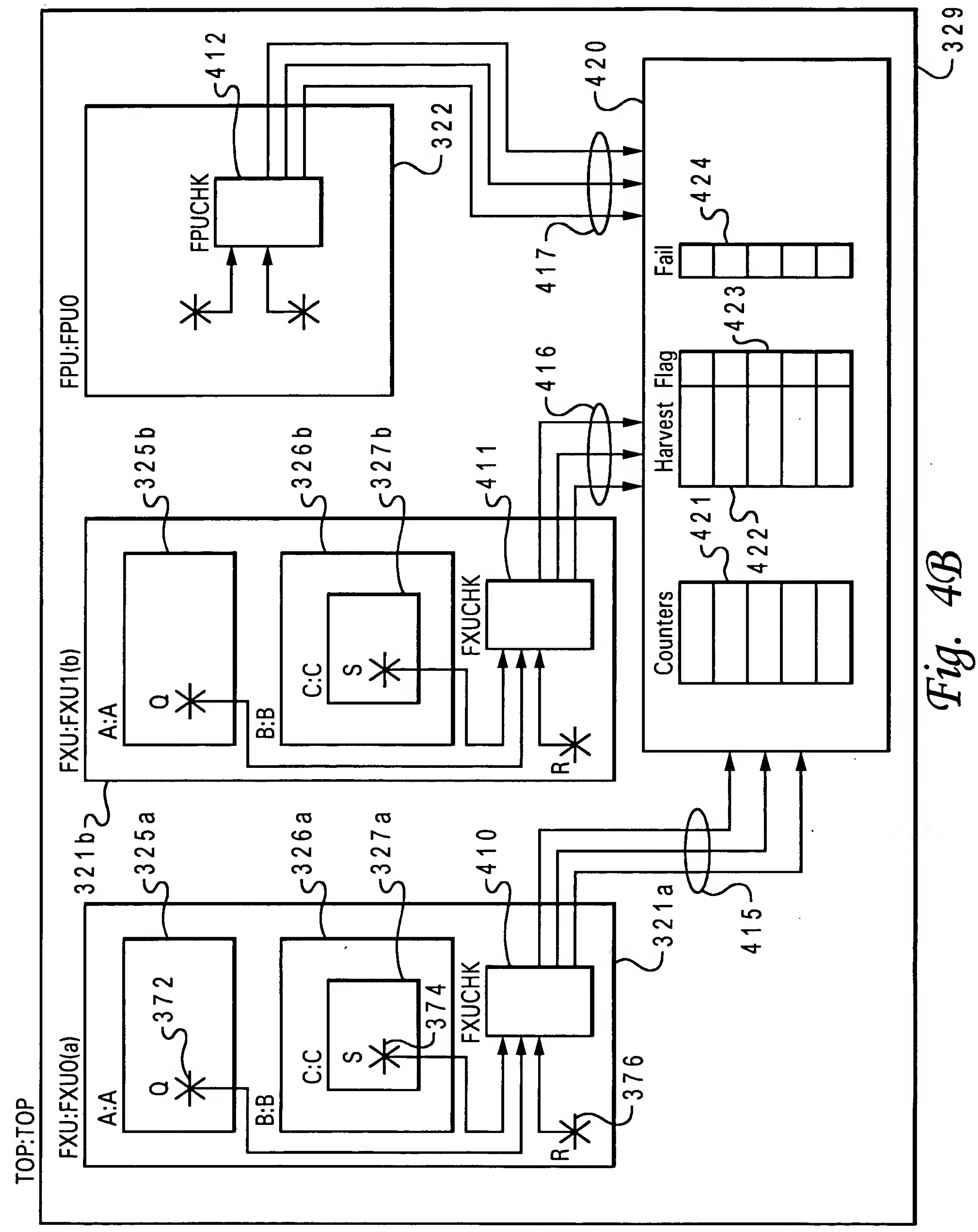
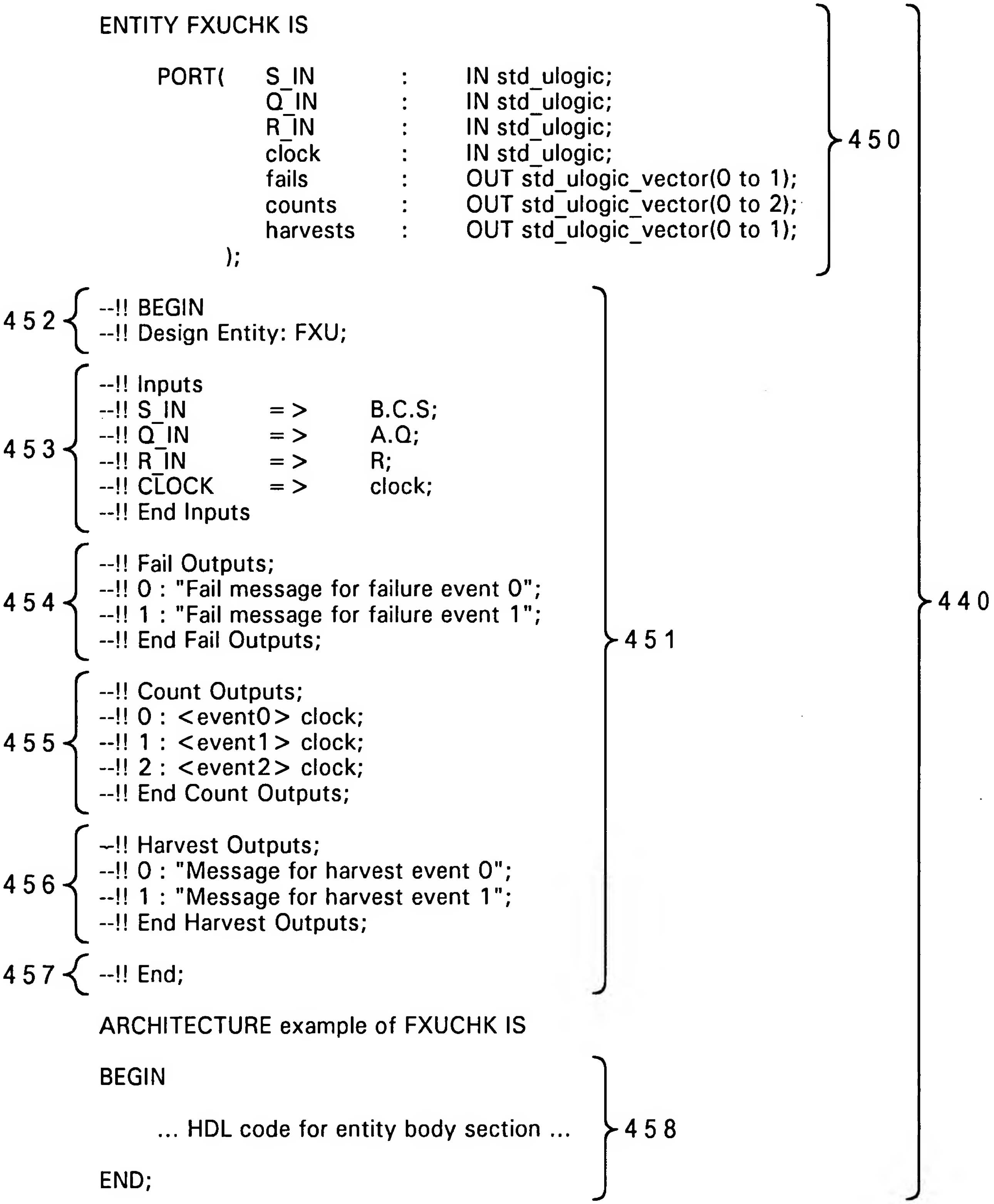

TOP:TOP
329
FXU:FXU0(a)
321a
A:A
B:B
C:C
Q
S
R
372
374
376
325a
326a
327a
FXUCHK
410
415
321b
FXU:FXU1(b)
A:A
B:B
C:C
Q
S
R
325b
326b
327b
FXUCHK
411
416
FPU:FPU0
322
FPUCHK
412
417
420
Counters
421
422
Harvest
Flag
423
Fail
424

Fig. 4B



```
ENTITY FXUCHK IS

      PORT(   S_IN       :     IN std_ulogic;
              Q_IN       :     IN std_ulogic;
              R_IN       :     IN std_ulogic;
              clock      :     IN std_ulogic;
              fails      :     OUT std_ulogic_vector(0 to 1);
              counts     :     OUT std_ulogic_vector(0 to 2);
              harvests   :     OUT std_ulogic_vector(0 to 1);
            );

--!! BEGIN
--!! Design Entity: FXU;

--!! Inputs
--!! S_IN       =>     B.C.S;
--!! Q_IN       =>     A.Q;
--!! R_IN       =>     R;
--!! CLOCK      =>     clock;
--!! End Inputs

--!! Fail Outputs;
--!! 0 : "Fail message for failure event 0";
--!! 1 : "Fail message for failure event 1";
--!! End Fail Outputs;

--!! Count Outputs;
--!! 0 : <event0> clock;
--!! 1 : <event1> clock;
--!! 2 : <event2> clock;
--!! End Count Outputs;

--!! Harvest Outputs;
--!! 0 : "Message for harvest event 0";
--!! 1 : "Message for harvest event 1";
--!! End Harvest Outputs;

--!! End;

ARCHITECTURE example of FXUCHK IS

BEGIN

      ... HDL code for entity body section ...

END;
```

Fig. 4C

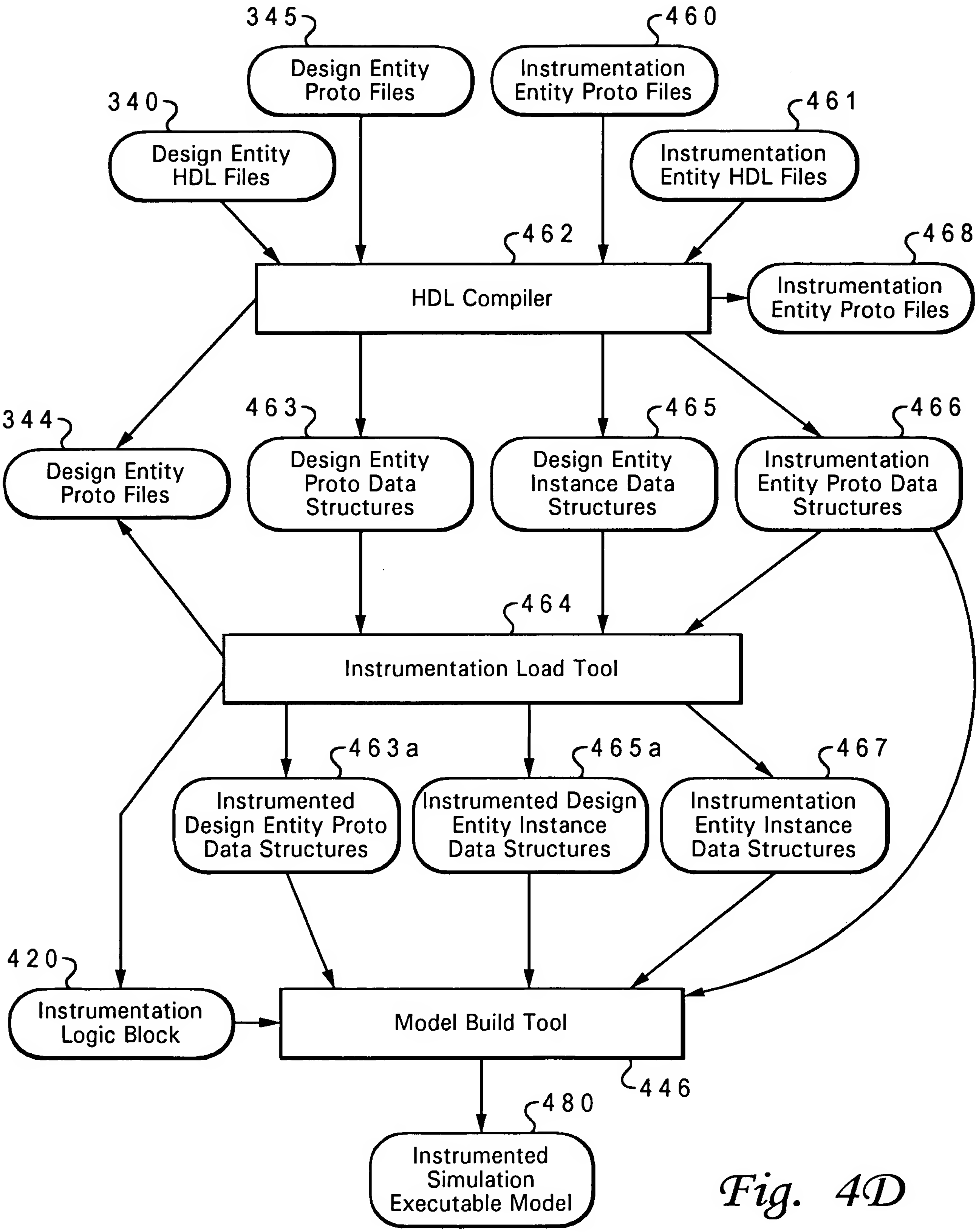

345
Design Entity Proto Files
460
Instrumentation Entity Proto Files
340
Design Entity HDL Files
461
Instrumentation Entity HDL Files
462
HDL Compiler
468
Instrumentation Entity Proto Files
344
Design Entity Proto Files
463
Design Entity Proto Data Structures
465
Design Entity Instance Data Structures
466
Instrumentation Entity Proto Data Structures
464
Instrumentation Load Tool
463a
Instrumented Design Entity Proto Data Structures
465a
Instrumented Design Entity Instance Data Structures
467
Instrumentation Entity Instance Data Structures
420
Instrumentation Logic Block
Model Build Tool
446
480
Instrumented Simulation Executable Model

Fig. 4D

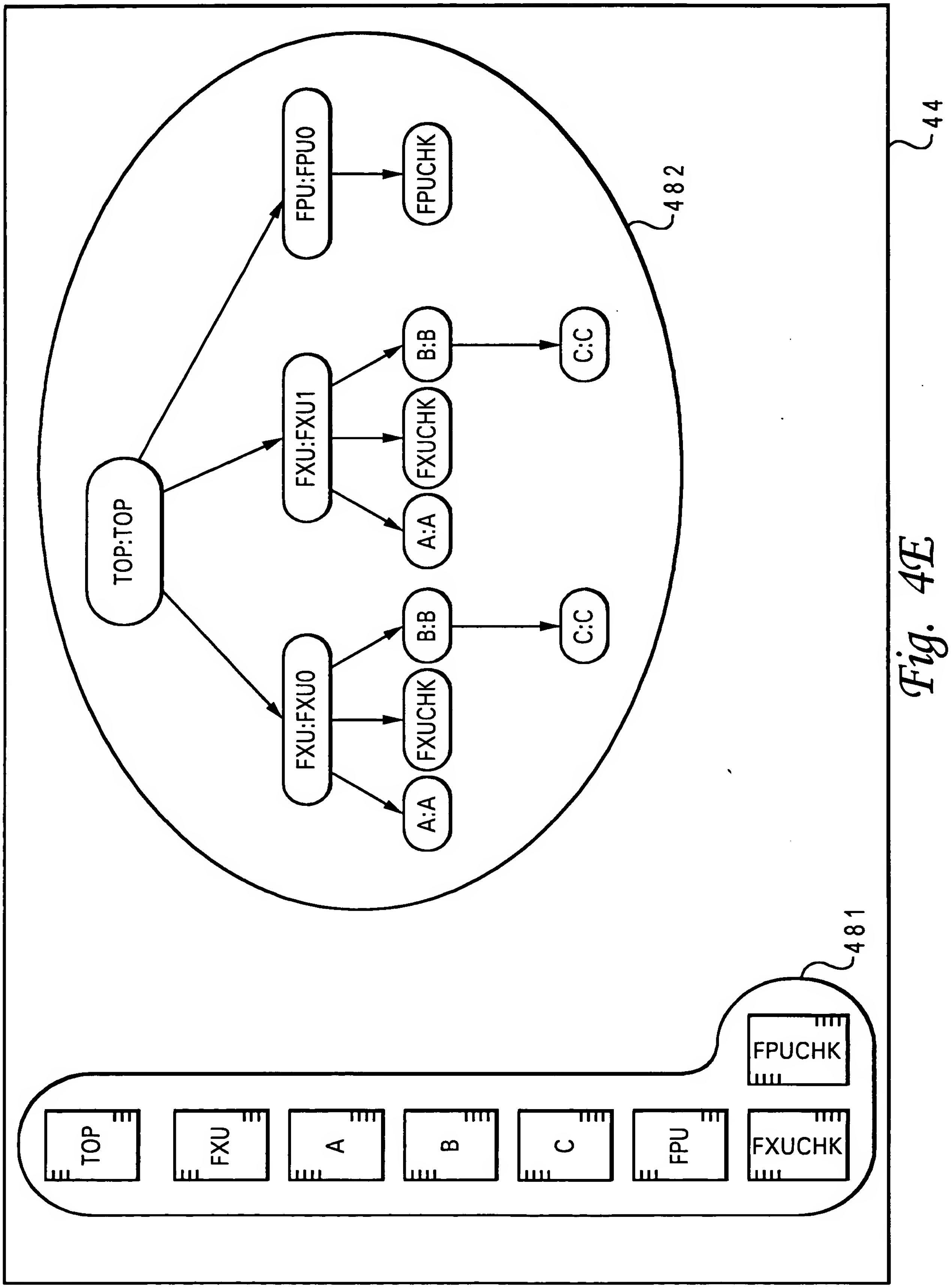

TOP:TOP
FXU:FXU0
FXU:FXU1
FPU:FPU0
A:A
FXUCHK
B:B
C:C
A:A
FXUCHK
B:B
C:C
FPUCHK
482
44
TOP
FXU
A
B
C
FPU
FXUCHK
FPUCHK
481

Fig. 4E

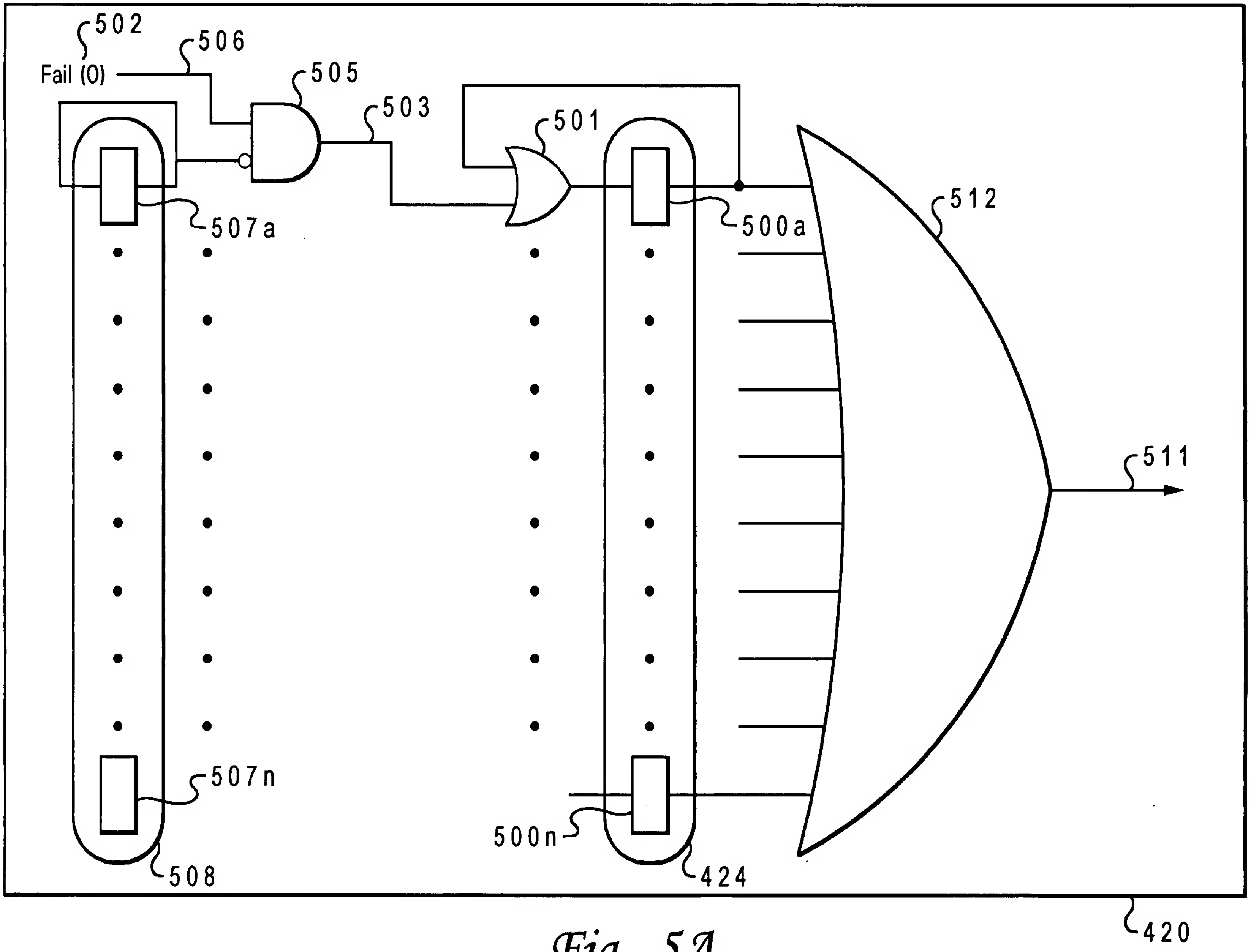

502
Fail (0)
506
505
503
501
512
507a
500a
511
507n
500n
508
424
420

Fig. 5A

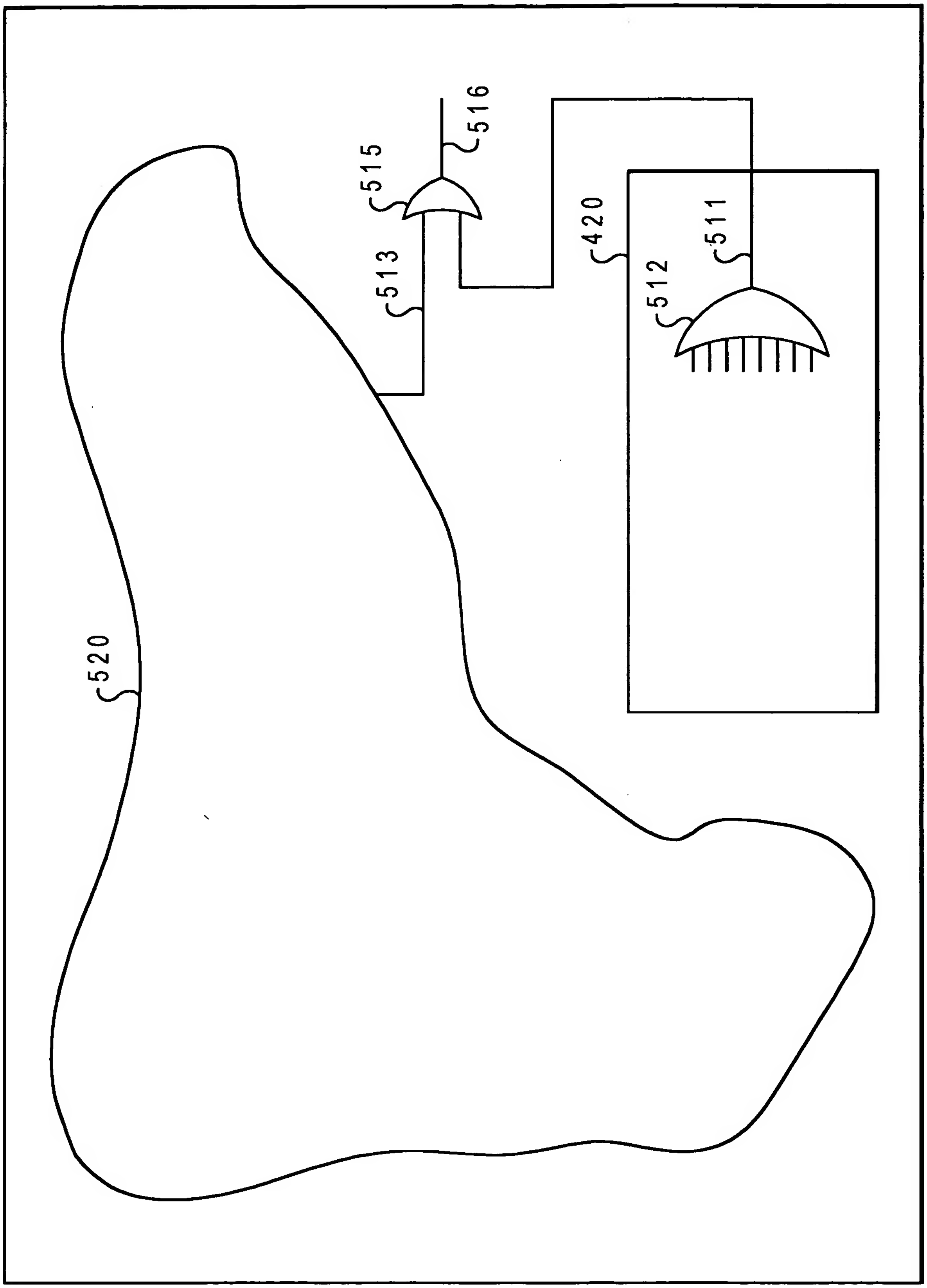


Fig. 5B

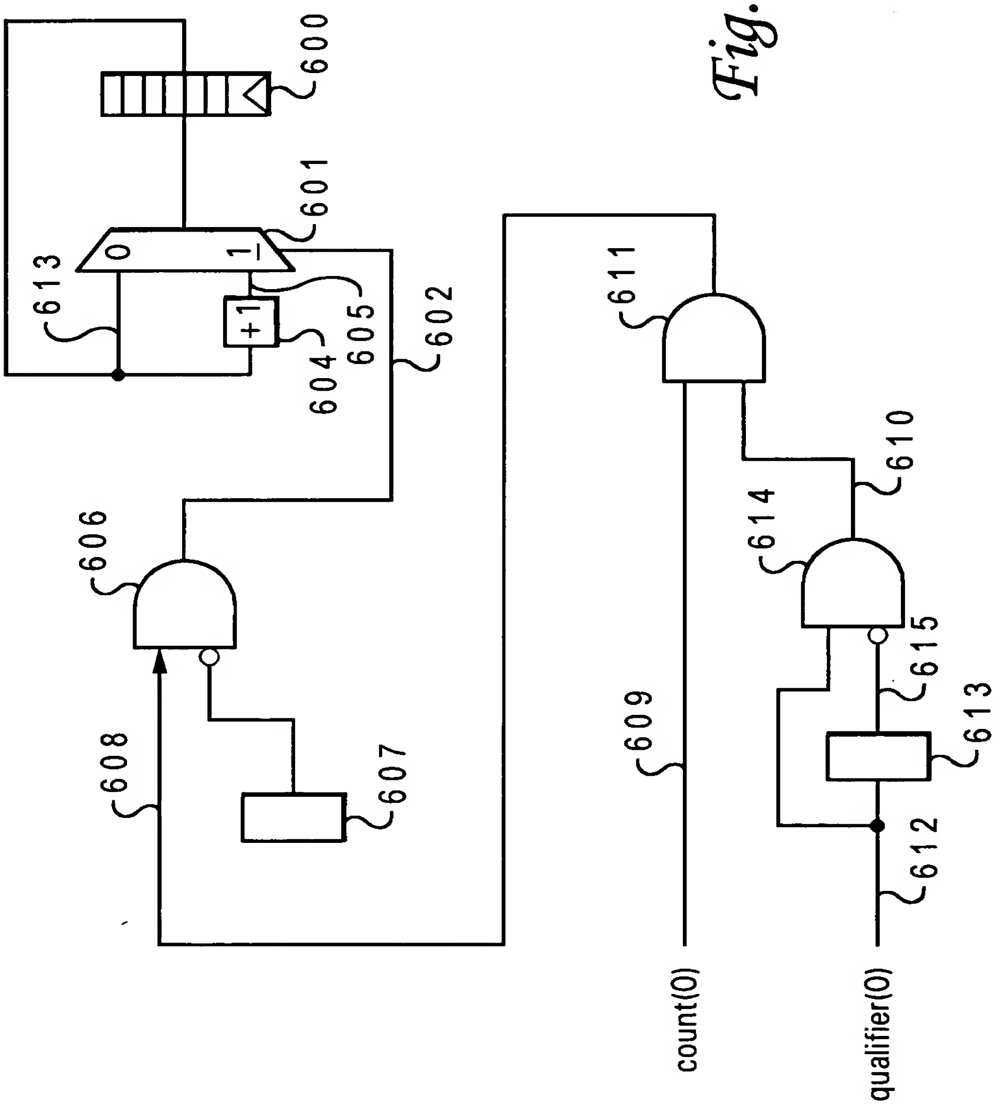

600
601
0
1
613
+1
605
604
602
606
608
607
611
610
614
615
613
609
612
count(0)
qualifier(0)

Fig. 6A

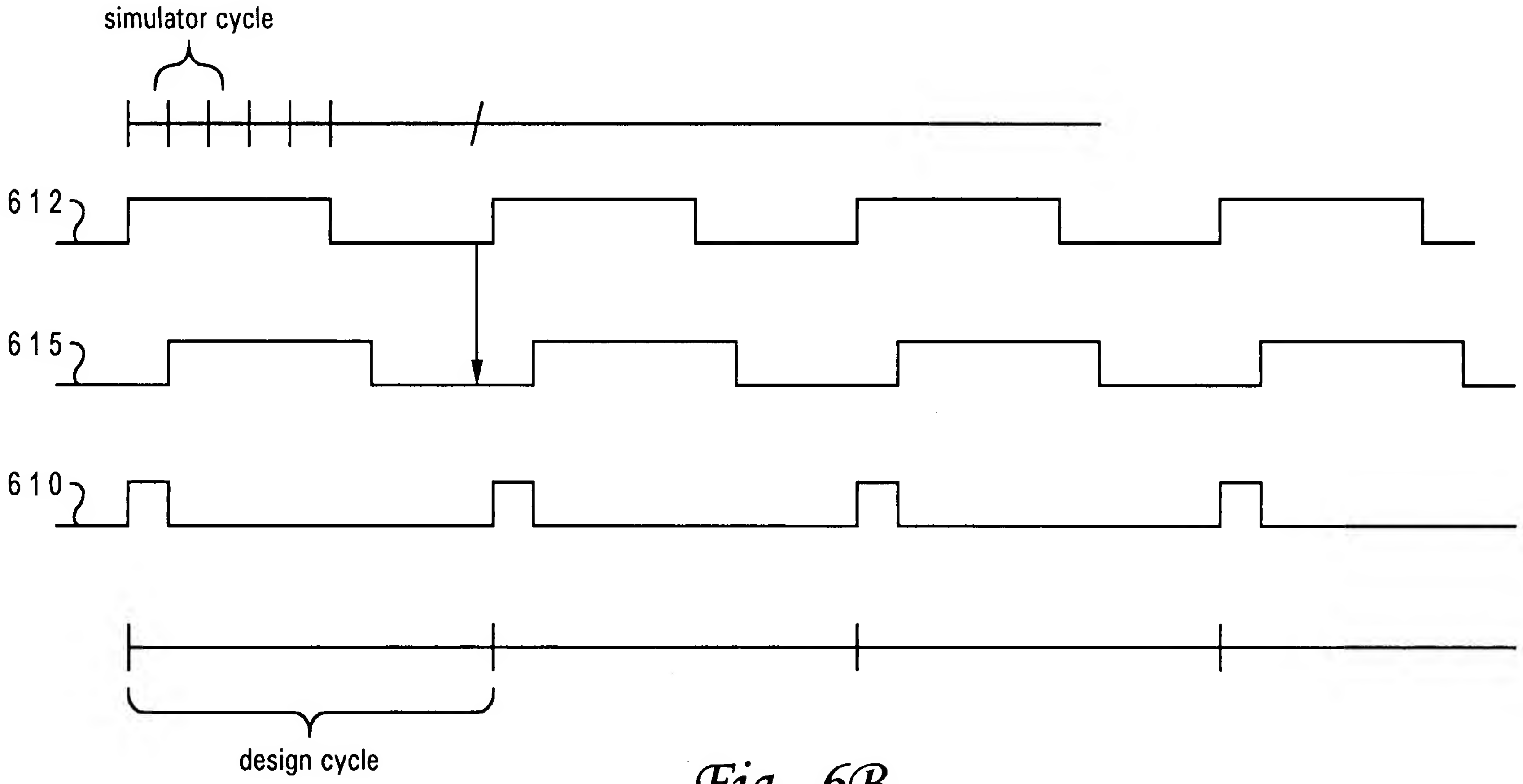


Fig. 6B

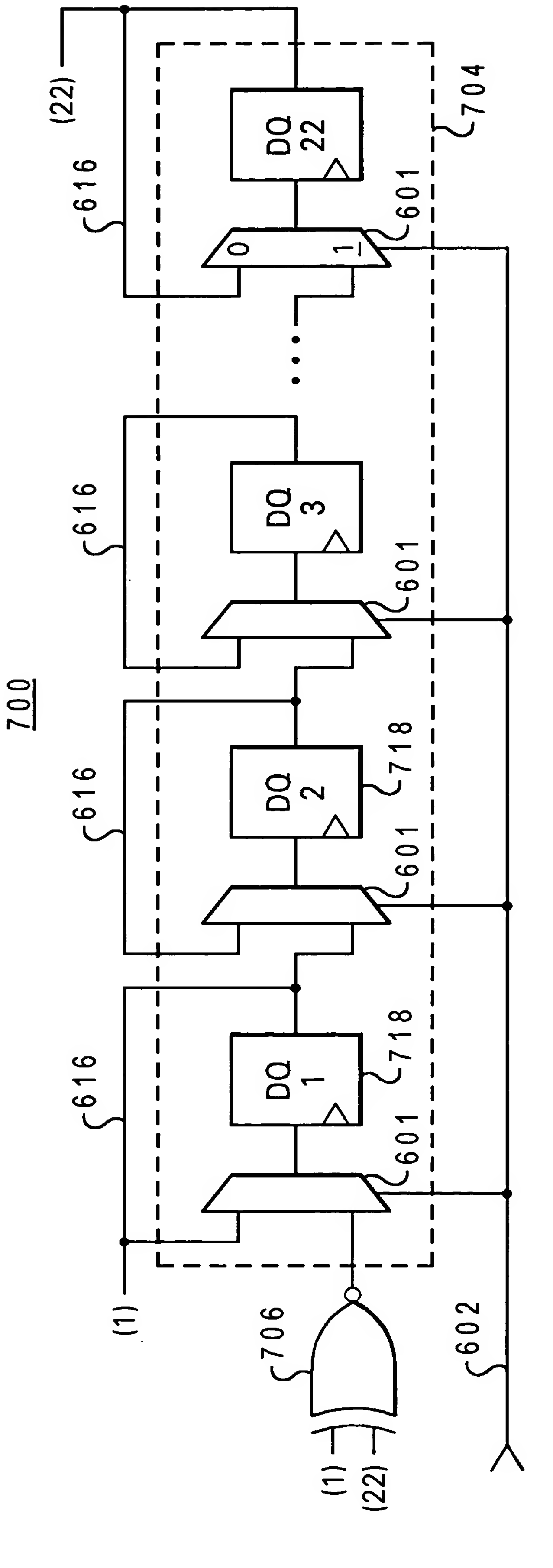

700
616
616
616
616
(22)
704
DQ
22
DQ
3
DQ
2
DQ
1
601
601
601
601
0
1
718
718
(1)
706
602
(1)
(22)

Fig. 7

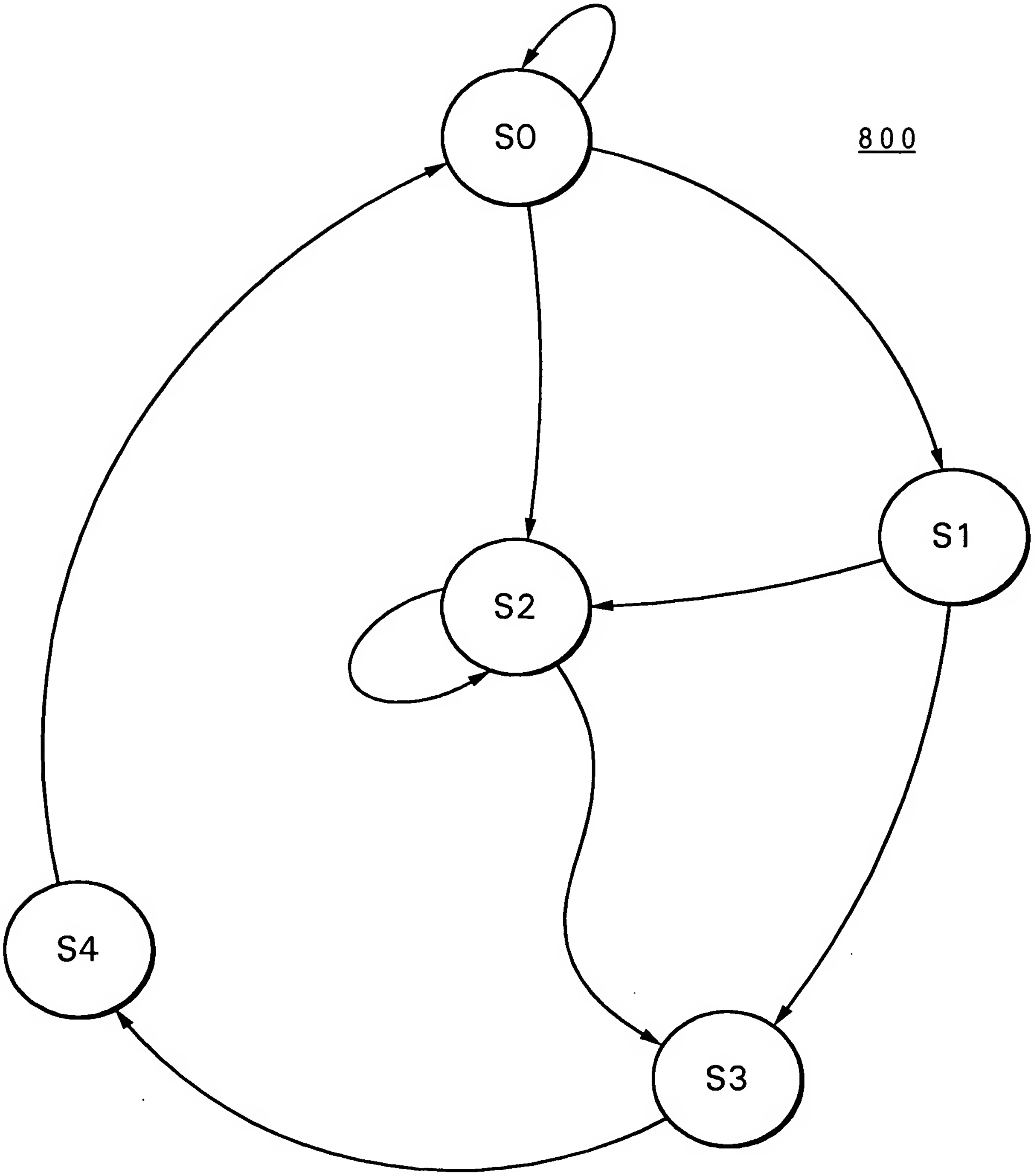


Fig. 8A

Prior Art

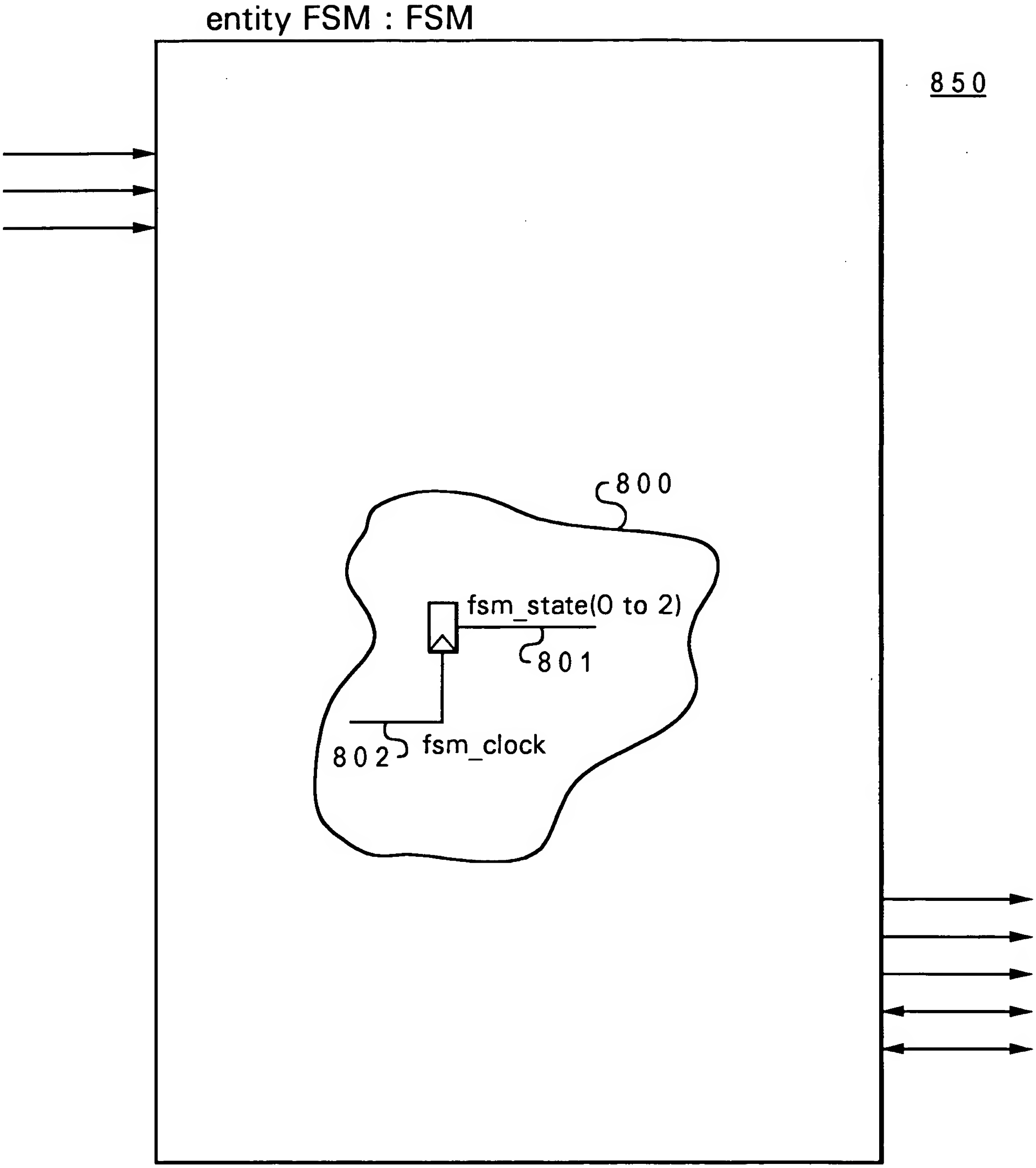
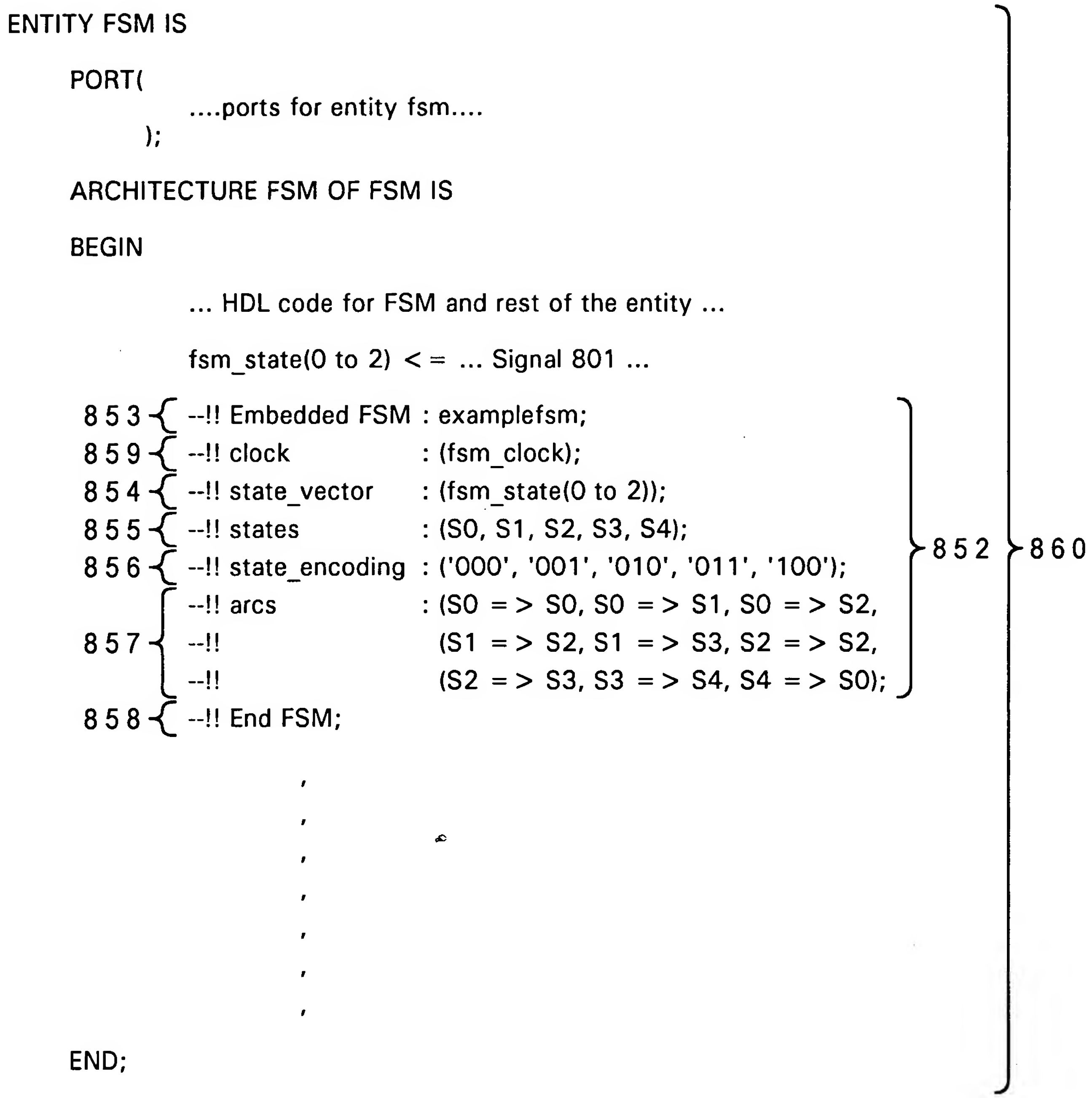


Fig. 8B

Prior Art



```
ENTITY FSM IS

     PORT(
                    ....ports for entity fsm....
            );

     ARCHITECTURE FSM OF FSM IS

     BEGIN

                    ... HDL code for FSM and rest of the entity ...

                    fsm_state(0 to 2) <= ... Signal 801 ...

                    --!! Embedded FSM  : examplefsm;
                    --!! clock         : (fsm_clock);
                    --!! state_vector  : (fsm_state(0 to 2));
                    --!! states        : (S0, S1, S2, S3, S4);
                    --!! state_encoding : ('000', '001', '010', '011', '100');
                    --!! arcs          : (S0 => S0, S0 => S1, S0 => S2,
                    --!!                 (S1 => S2, S1 => S3, S2 => S2,
                    --!!                 (S2 => S3, S3 => S4, S4 => S0);
                    --!! End FSM;
                              ,
                              ,
                              ,
                              ,
                              ,
                              ,
                              ,

     END;
```

Fig. 8C

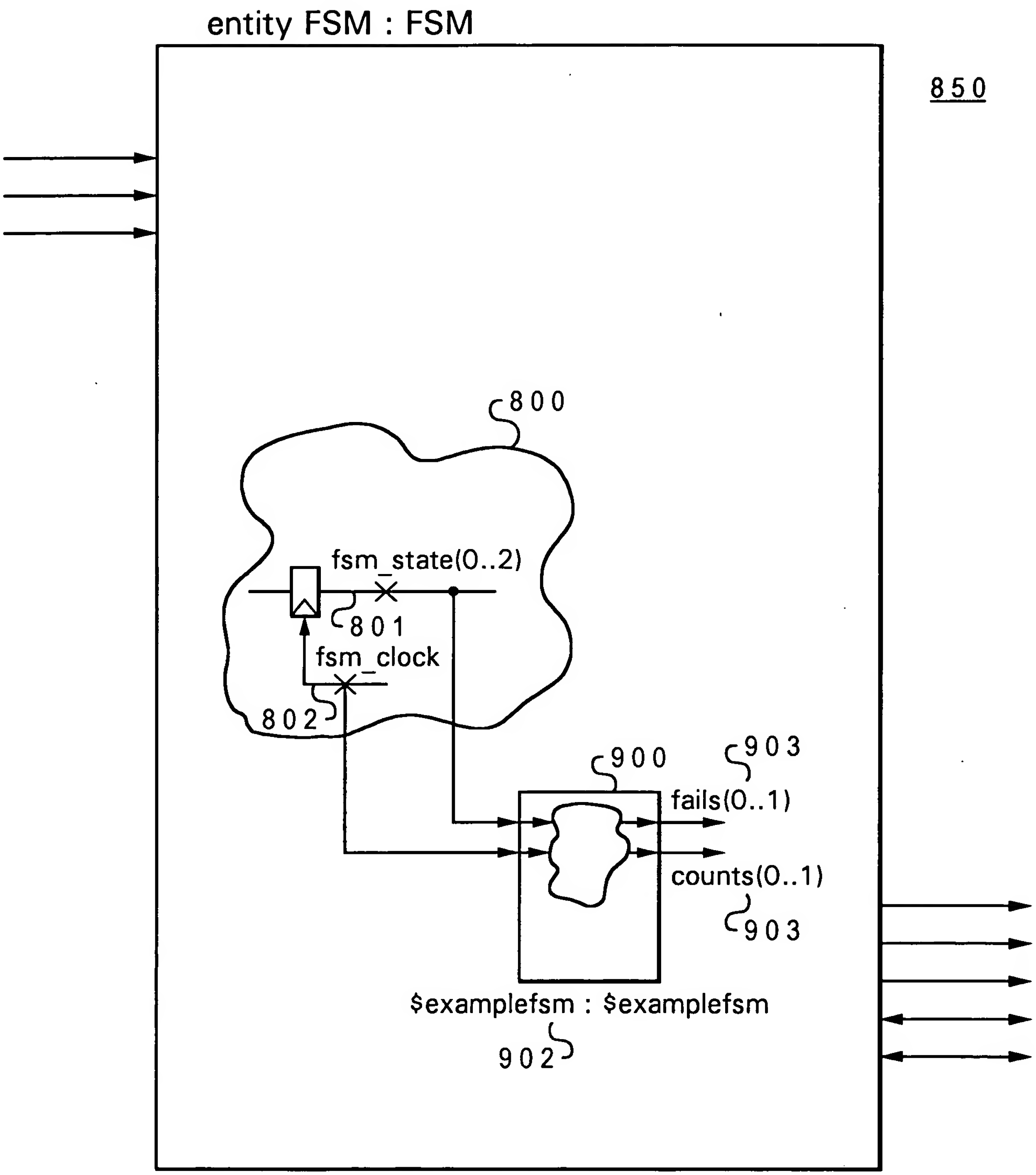

entity FSM : FSM
850
800
fsm_state(0..2)
801
fsm_clock
802
900
903
fails(0..1)
counts(0..1)
903
$examplefsm : $examplefsm
902

Fig. 9

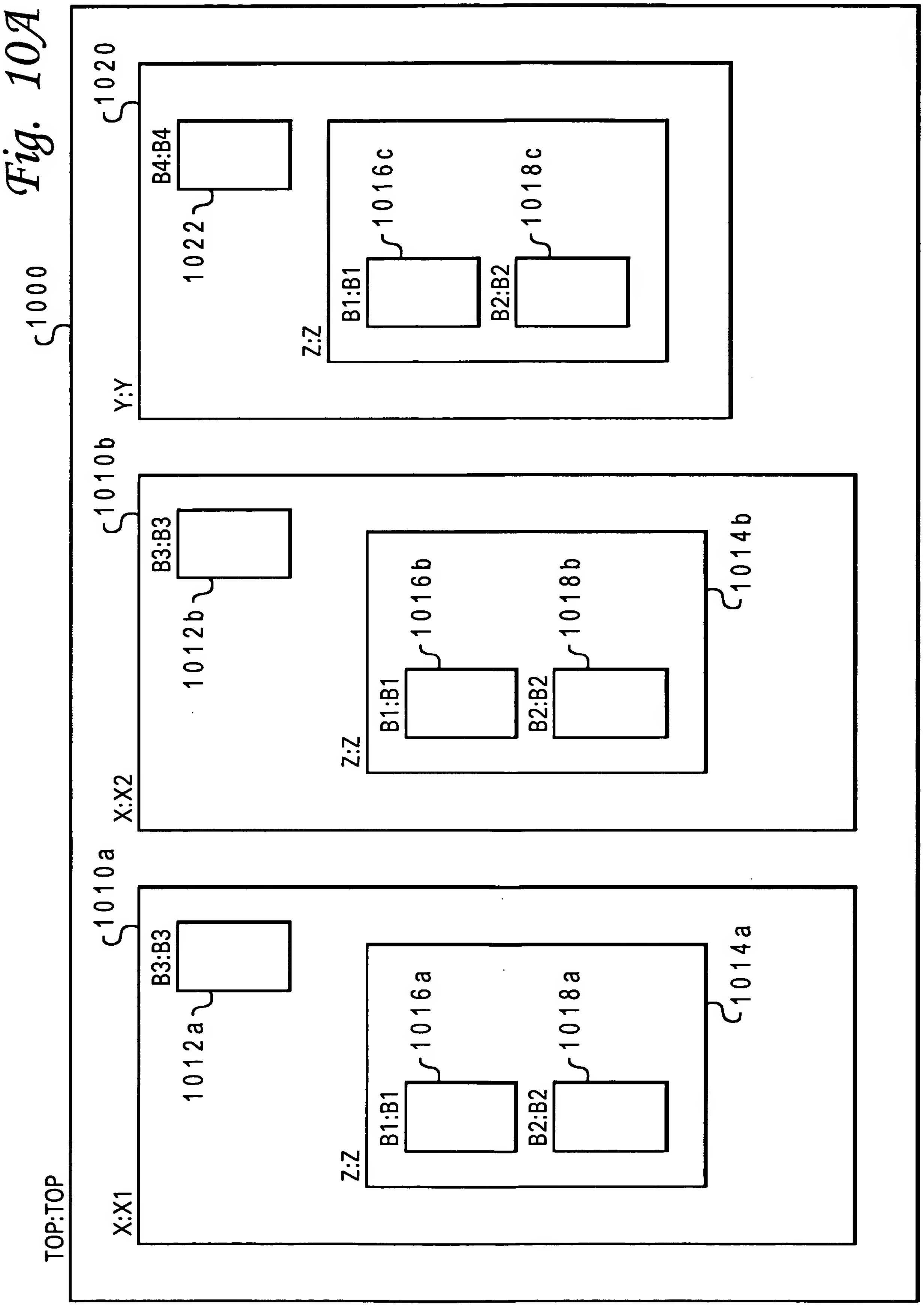
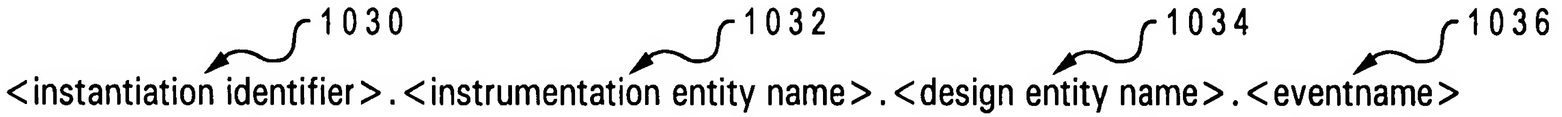
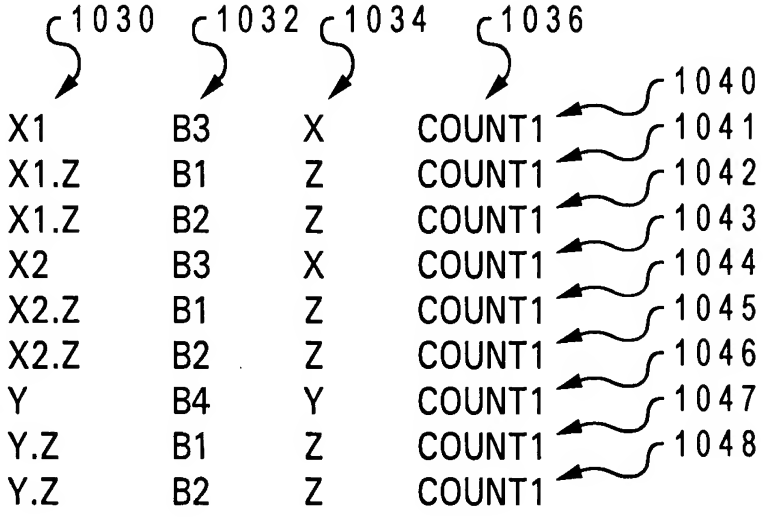

TOP:TOP
1000
X:X1
1010a
B3:B3
1012a
Z:Z
1014a
B1:B1
1016a
B2:B2
1018a
X:X2
1010b
B3:B3
1012b
Z:Z
1014b
B1:B1
1016b
B2:B2
1018b
Y:Y
1020
B4:B4
1022
Z:Z
B1:B1
1016c
B2:B2
1018c

Fig. 10A



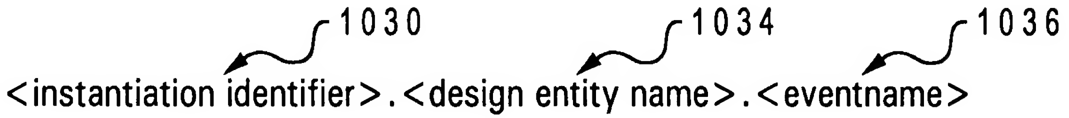
<instantiation identifier>.<instrumentation entity name>.<design entity name>.<eventname>

Fig. 10B



X1	B3	X	COUNT1
X1.Z	B1	Z	COUNT1
X1.Z	B2	Z	COUNT1
X2	B3	X	COUNT1
X2.Z	B1	Z	COUNT1
X2.Z	B2	Z	COUNT1
Y	B4	Y	COUNT1
Y.Z	B1	Z	COUNT1
Y.Z	B2	Z	COUNT1

Fig. 10C



<instantiation identifier>.<design entity name>.<eventname>

Fig. 10D

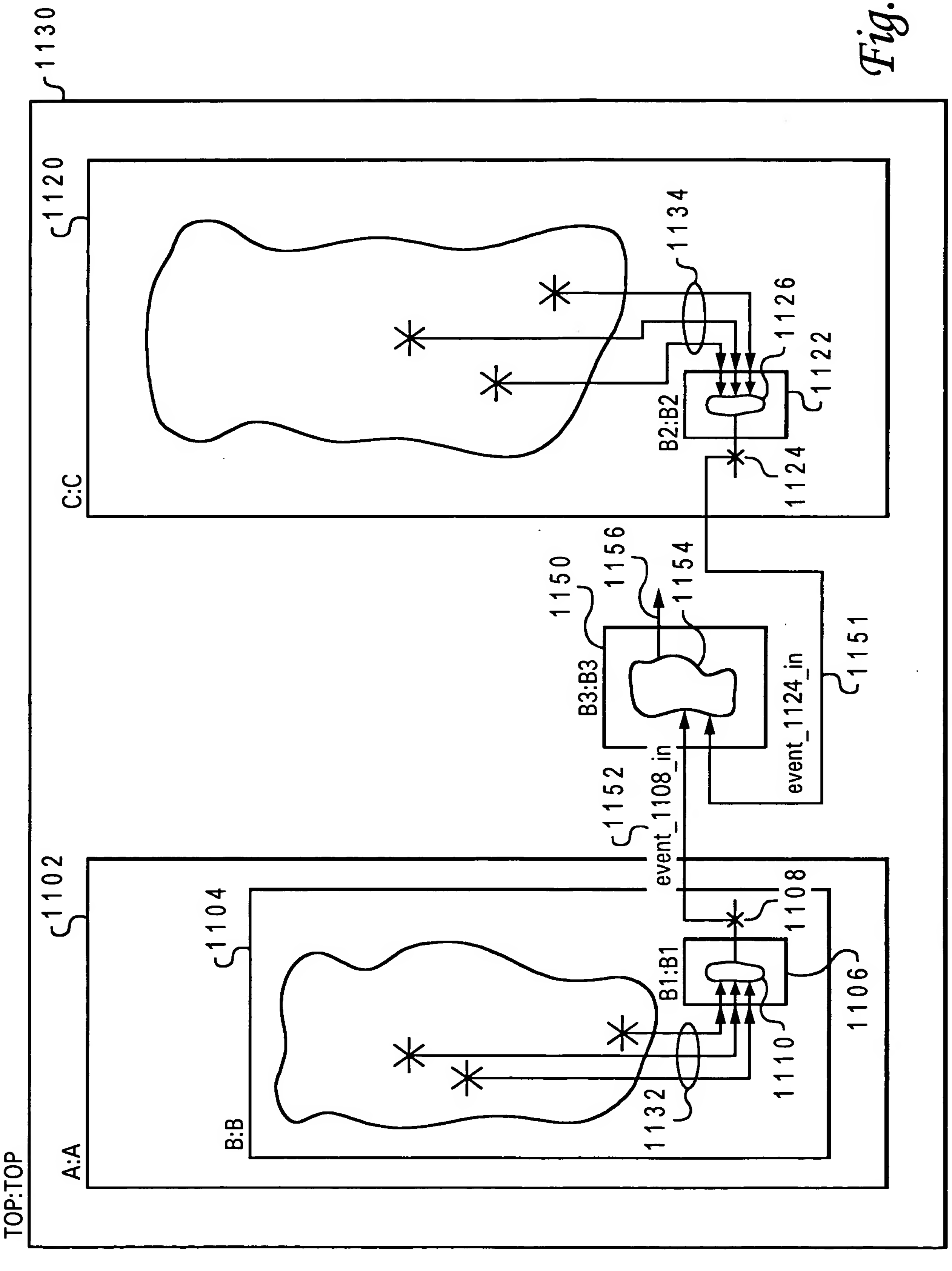
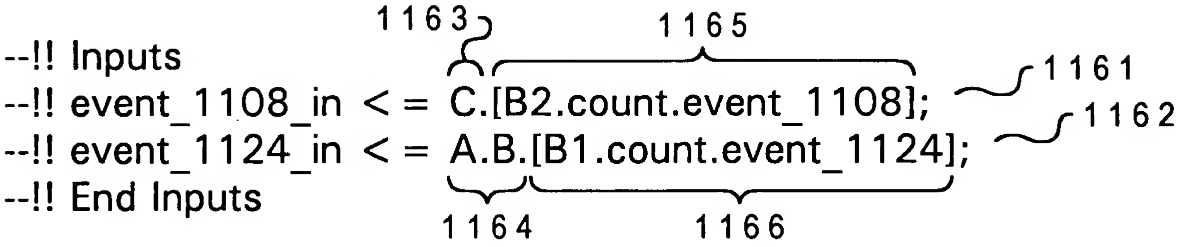
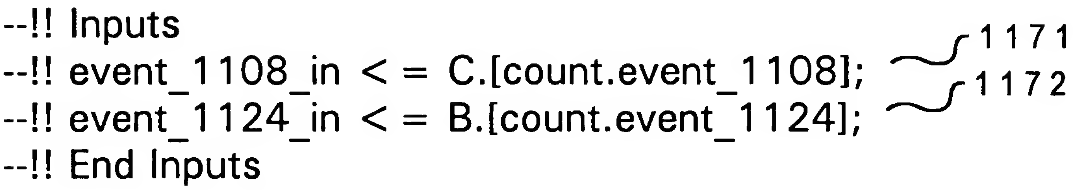

TOP:TOP
1130
A:A
1102
B:B
1104
B1:B1
1106
1108
1110
1132
event_1108_in
1152
B3:B3
1150
1156
1154
event_1124_in
1151
C:C
1120
1134
1126
1122
B2:B2
1124

Fig. 11A



```
--!! Inputs
--!! event_1108_in <= C.[B2.count.event_1108];
--!! event_1124_in <= A.B.[B1.count.event_1124];
--!! End Inputs
```

Fig. 11B



```
--!! Inputs
--!! event_1108_in <= C.[count.event_1108];
--!! event_1124_in <= B.[count.event_1124];
--!! End Inputs
```

Fig. 11C

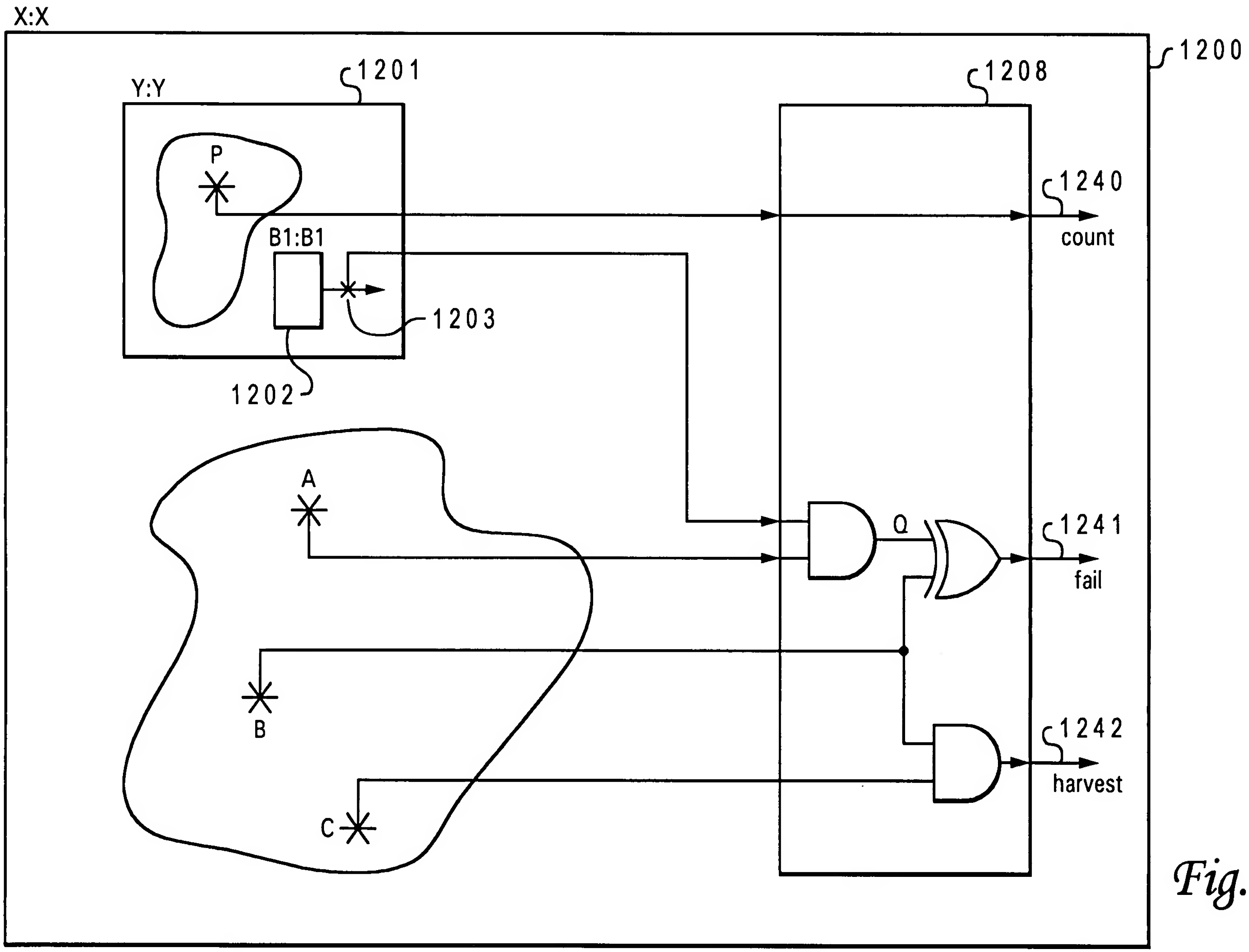
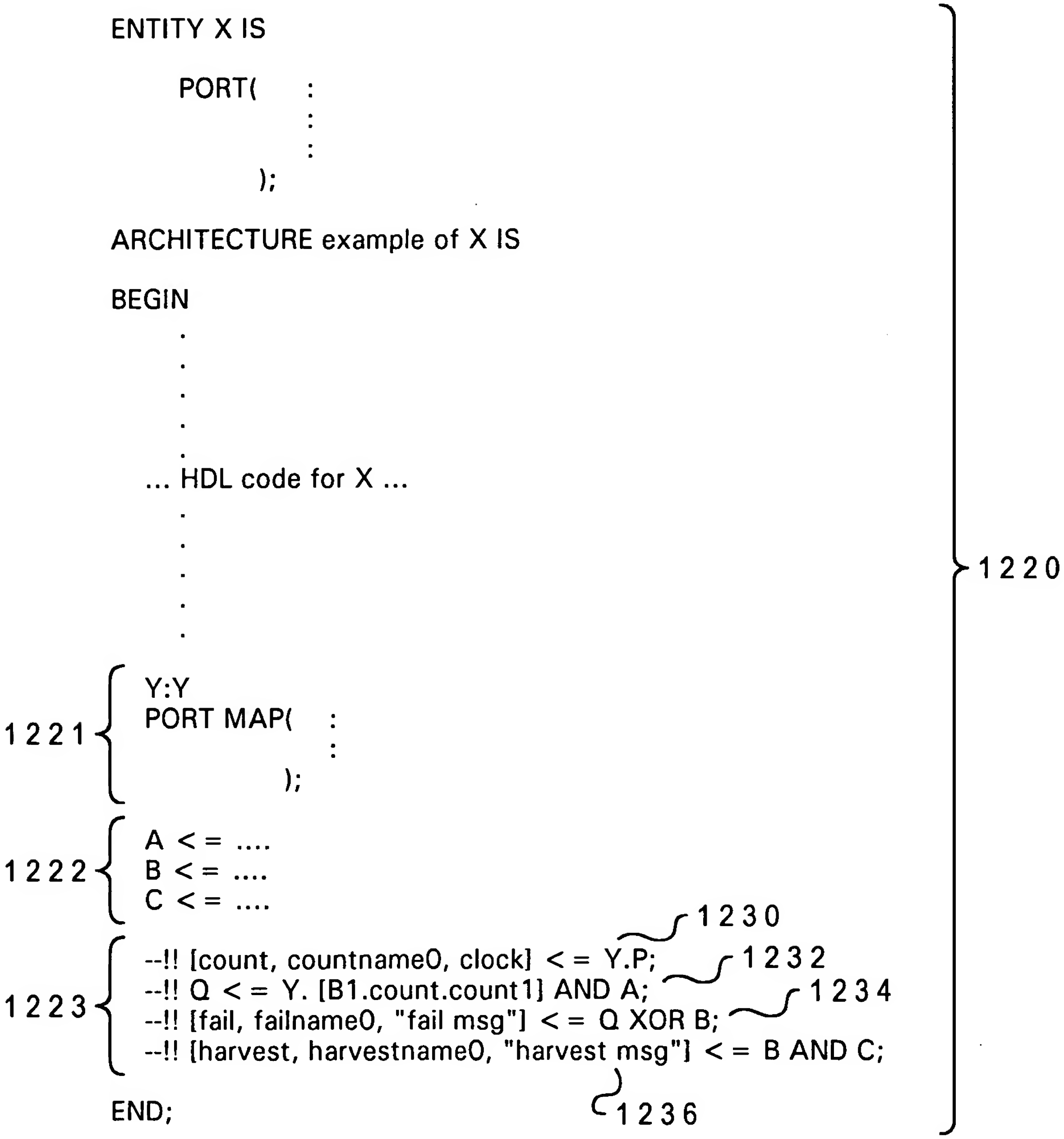

X:X
1200
Y:Y
1201
P
B1:B1
1203
1202
1208
1240
count
A
Q
1241
fail
B
1242
harvest
C

Fig. 12A



```
ENTITY X IS

       PORT(      :
                  :
                  :
              );

ARCHITECTURE example of X IS

BEGIN
       .
       .
       .
       .
       .
   ... HDL code for X ...
       .
       .
       .
       .
       .

   Y:Y
   PORT MAP(     :
                 :
             );

   A <= ....
   B <= ....
   C <= ....

   --!! [count, countname0, clock] < = Y.P;
   --!! Q < = Y. [B1.count.count1] AND A;
   --!! [fail, failname0, "fail msg"] < = Q XOR B;
   --!! [harvest, harvestname0, "harvest msg"] < = B AND C;

END;
```

Fig. 12B

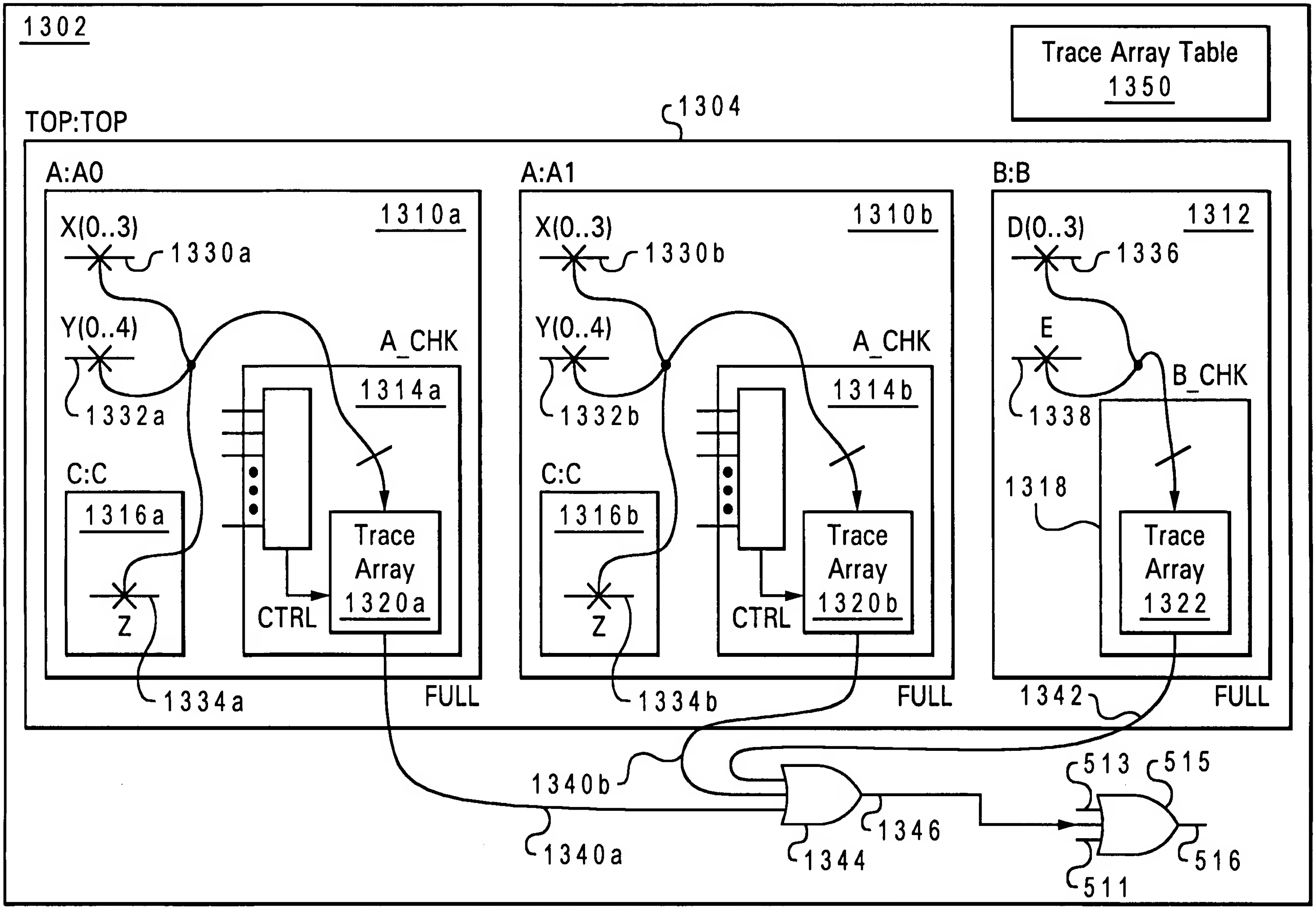

1302
Trace Array Table
1350
TOP:TOP
1304
A:A0
1310a
X(0..3)
1330a
Y(0..4)
1332a
A_CHK
1314a
C:C
1316a
Z
CTRL
Trace
Array
1320a
1334a
FULL
A:A1
1310b
X(0..3)
1330b
Y(0..4)
1332b
A_CHK
1314b
C:C
1316b
Z
CTRL
Trace
Array
1320b
1334b
FULL
B:B
1312
D(0..3)
1336
E
1338
B_CHK
1318
Trace
Array
1322
1342
FULL
1340b
1340a
1344
1346
513
515
511
516

Fig. 13

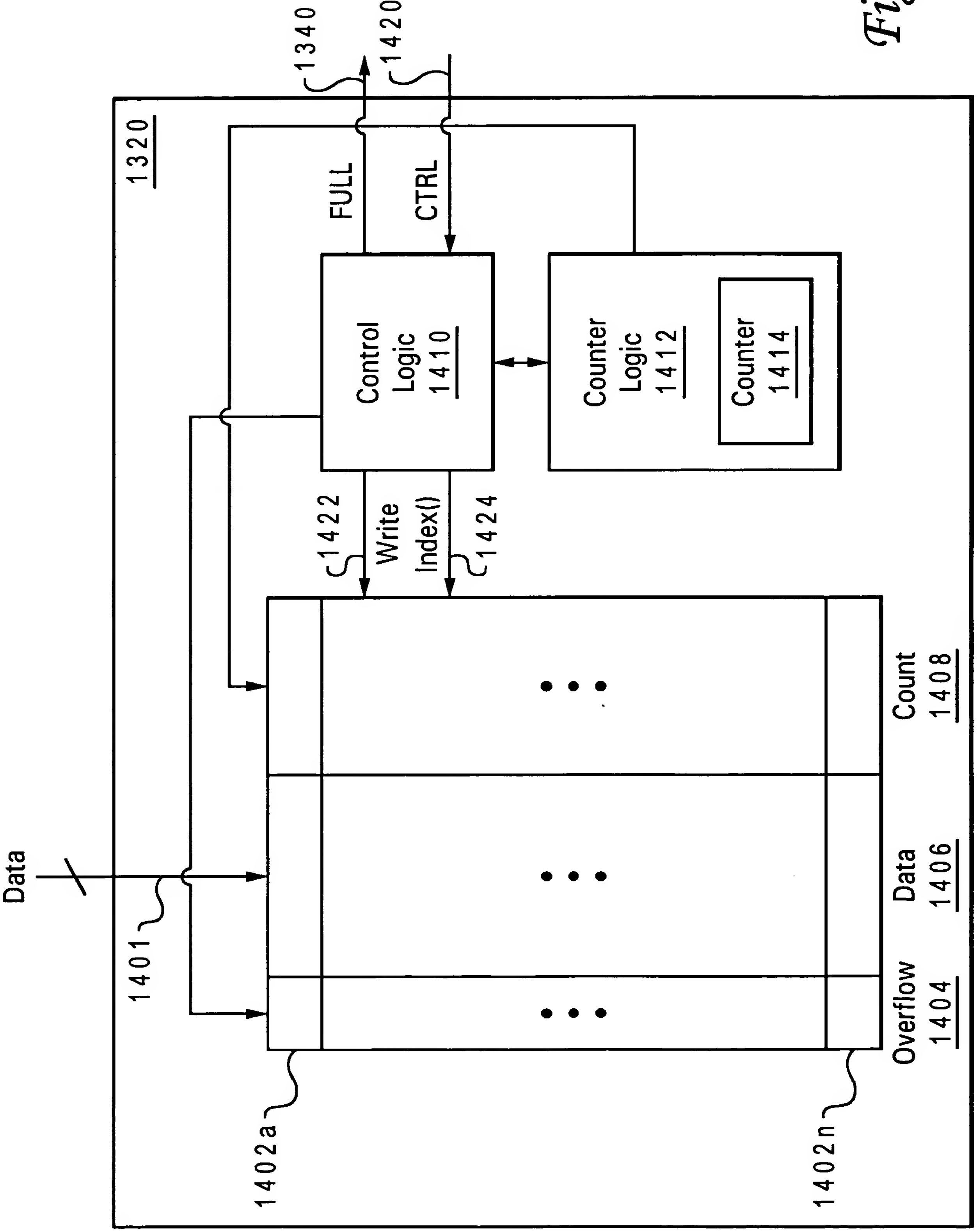

1320
1340
1420
FULL
CTRL
Control Logic 1410
Counter Logic 1412
Counter 1414
1422
Write
Index()
1424
Data
1401
Count 1408
Data 1406
Overflow 1404
1402a
1402n

Fig. 14

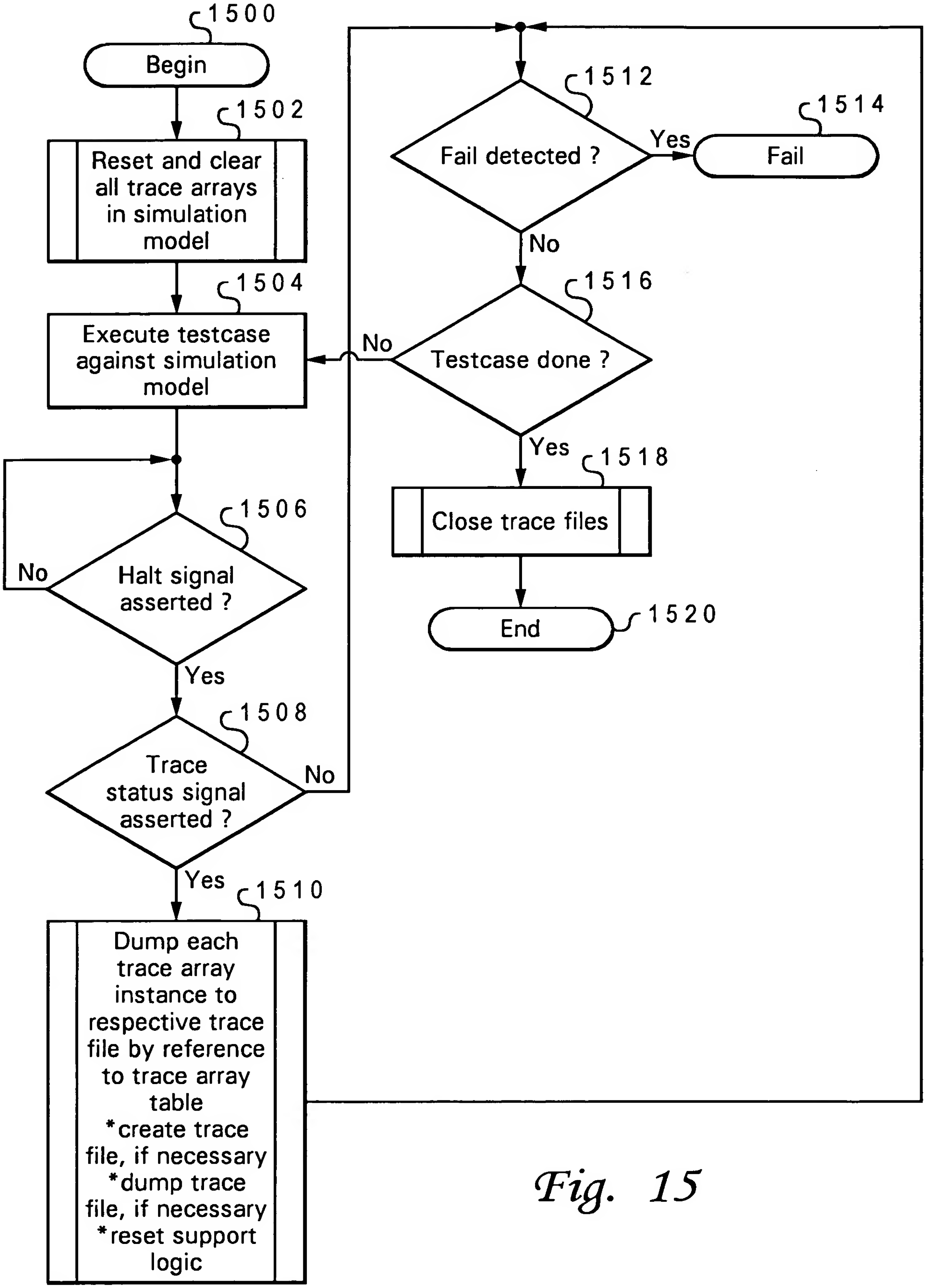

1500
Begin
1502
Reset and clear all trace arrays in simulation model
1504
Execute testcase against simulation model
1506
Halt signal asserted ?
No
Yes
1508
Trace status signal asserted ?
No
Yes
1510
Dump each trace array instance to respective trace file by reference to trace array table
*create trace file, if necessary
*dump trace file, if necessary
*reset support logic
1512
Fail detected ?
Yes
1514
Fail
No
1516
Testcase done ?
No
Yes
1518
Close trace files
End
1520

Fig. 15

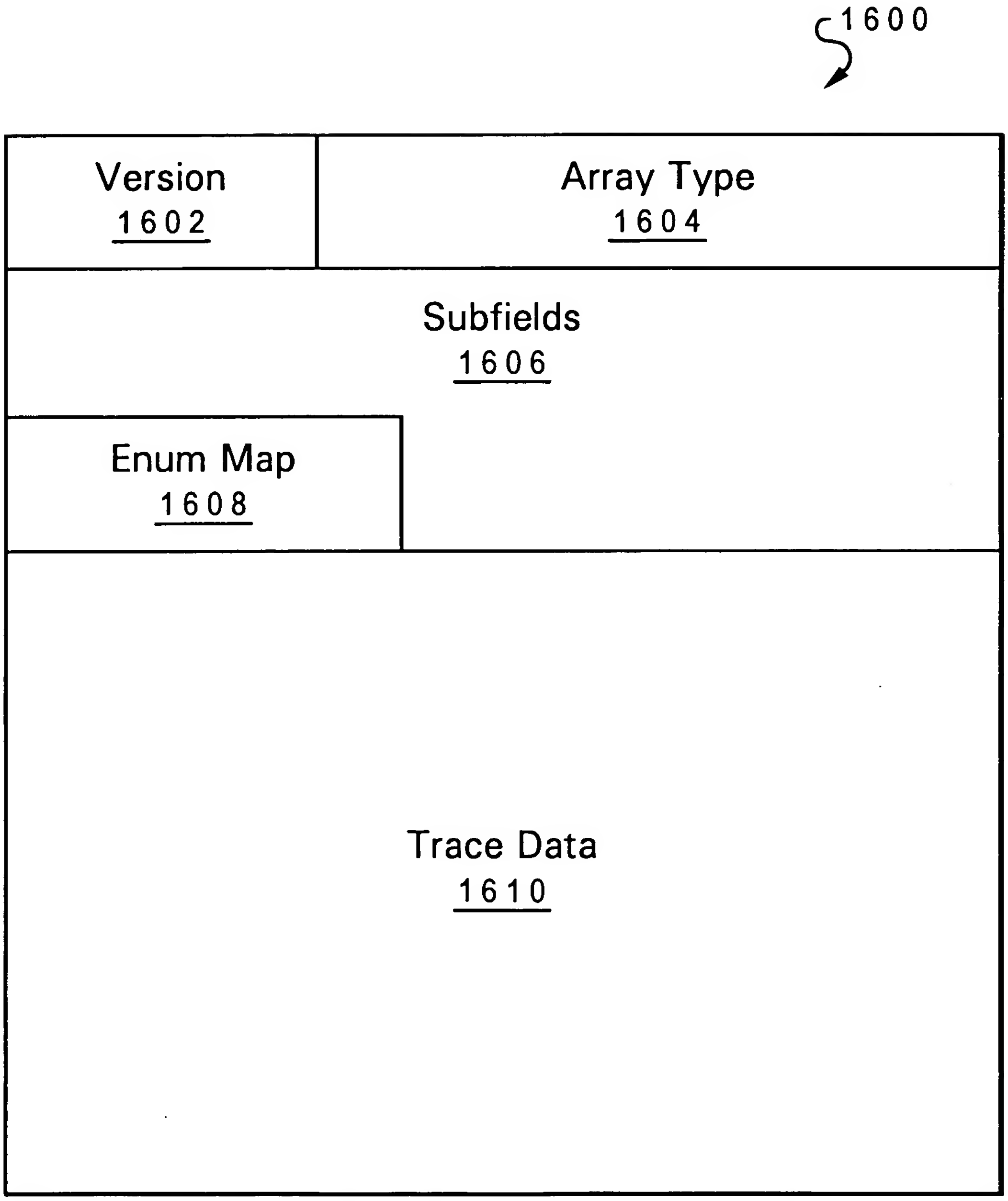

1600
Version
1602
Array Type
1604
Subfields
1606
Enum Map
1608
Trace Data
1610

Fig. 16

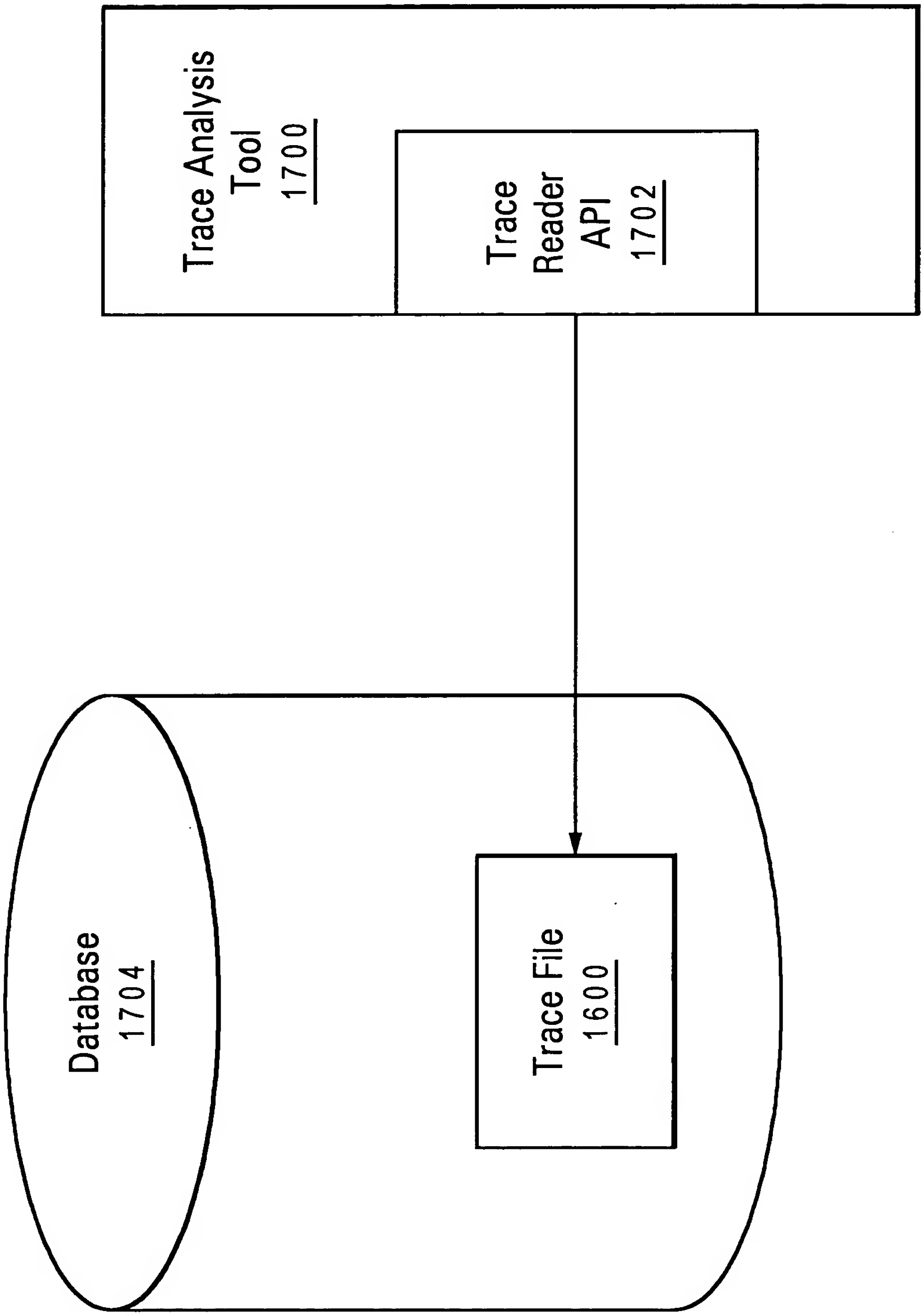

Trace Analysis Tool 1700
Trace Reader API 1702
Database 1704
Trace File 1600

Fig. 17